

# THÈSE

présentée devant

**L'UNIVERSITÉ DE VERSAILLES-SAINT-QUENTIN-EN-YVELINES**

le 23 Octobre 2003

pour l'obtention

**du DIPLÔME DE DOCTORAT**

(arrêté du 30.3.92)

spécialité : INFORMATIQUE

par

**Jérôme Grosjean**

**ENVIRONNEMENTS VIRTUELS : CONTRÔLE D'APPLICATION  
ET EXPLORATION DE SCÈNES 3D**

Rapporteurs : Pere Brunet  
Pascal Guitton  
Yvan Poupyrev

Examinateurs : Sabine Coquillart  
Philippe Fuchs  
Nacer M'sirdi  
Claude Puech



*Tout ce qu'on invente est vrai*

*Gustave Flaubert*



## Résumé

Les configurations de réalité virtuelle à base de projection sur grands écrans proposent des environnements virtuels (EV) immersifs, où l'utilisateur peut interagir directement sur des objets 3D, pour les manipuler, les transformer ou pour naviguer au sein de la scène. Les techniques d'interaction pour ces environnements sont encore jeunes et prospectives et des besoins cruciaux pour le développement industriel d'applications pour EV existent.

Une présentation des différentes techniques existantes à ce jour pour les trois grandes classes de tâches génériques de l'interaction en environnement virtuel : la navigation, la sélection/manipulation et le contrôle d'application, montre les difficultés de l'interaction en milieu 3D immersif et les besoins des applications pour environnement virtuel.

Un premier frein à leur développement est l'absence de système commun et abouti de contrôle d'applications pour ces environnements. Dans les environnements à deux dimensions (station de travail), des métaphores existent pour contrôler les applications. On peut notamment citer la métaphore du bureau (avec fenêtres, menus déroulants, etc.) qui s'est largement imposée comme méthode principale de communication entre l'utilisateur et les programmes qu'il utilise. Dans les environnements virtuels immersifs à trois dimensions, il n'existe pas encore de consensus sur la meilleure approche à utiliser pour réaliser le contrôle d'applications.

Dans ce contexte, nous avons cherché à développer une nouvelle métaphore, s'appuyant sur un espace à trois dimensions. Nous nous sommes penchés dans un premier temps sur la réalisation d'un menu simplifié, réduit à un petit jeu de commandes, mais à accès rapide. Cette approche poursuit les mêmes objectifs que le mécanisme des raccourcis clavier, bien connu des utilisateurs de stations de travail. Le menu développé, baptisé Command & Control Cube (d'abréviation  $C^3$ ) est une interface entièrement conçue pour un univers 3D. Evolutive et légère, elle permet d'émettre rapidement des commandes vers une application grâce à une structure intuitive en forme de cube et une manipulation par brefs mouvements de la main dans des directions fixes. Un paradigme commun permet une manipulation pour utilisateur novice à l'aide de l'interface graphique, ou pour utilisateur expert sans retour visuel par mouvements rapides et "en aveugle" dans l'espace. Après un série d'évaluations formelles, ayant validé le principe du  $C^3$ , celui-ci a été étendu à une version hiérarchique, puis contextuelle, recréant ainsi pour les univers virtuels immersifs l'ensemble des grandes techniques de contrôle des menus déroulants 2D.

Dans le même contexte, les environnements virtuels immersifs manquent de certains des outils fondamentaux dont disposent couramment les stations de travail du monde 2D.

Ainsi, il n'existe encore à l'heure actuelle que très peu de méthodes pour saisir du texte ou des données numériques dans ces environnements. Principalement, la nature immersive des configurations d'environnement virtuel prohibe l'usage du clavier traditionnel ou de substituts trop encombrants, en terme de poids comme d'occupation de l'espace . Une méthode a été mise au point lors de la thèse, visant à fournir à l'utilisateur une méthode simple et légère de saisie de texte. Elle s'inspire d'une technique pour PDA (*Personal Digital Assistant*) nommée QuikWrite.

Enfin, un outil multifonctionnel d'exploration de scènes 3D a été mis au point. Une palette virtuelle tenue en main présente une surface rectangulaire pouvant être utilisée interactivement par l'utilisateur pour observer la scène qui l'entoure à l'aide de filtres. Ces filtres peuvent réaliser toutes sortes d'opérations : effet miroir, grossissement, éclairage altéré, coupe d'un modèle, etc. Une fonction particulière du mode réfléchissant permet de retrouver des objets perdus hors du champ de vision, sur des configurations semi-immersives comme le Plan de Travail Virtuel qui n'entourent pas complètement l'utilisateur d'un univers virtuel.

## Abstract

The virtual reality configurations based on large screen displays create immersive virtual environments (VE) where a user can interact directly with the 3D objects, to manipulate them, transform them or to navigate in the scene. The interaction techniques for these environments are still young and prospective. Crucial needs for the industrial development of VE applications exist.

A presentation of the different current techniques for the three main interaction tasks in virtual environments : navigation, selection/manipulation and application control shows the difficulties of interacting in a 3D immersive environment and the needs of these applications.

The absence of a universal and successful control application system for these environments is a first brake to their development. In the 2D environments (workstations), metaphors exist to control the applications. One can cite the desktop metaphor (with windows, pop-up menus, etc.) which stood out as the main communication method between the user and the applications. In 3D virtual environments, there is still no consensus on the best approach to control applications, due to the youth of this research field. In this context we focused first on the creation of a simplified menu, reduced to a small set of commands, but with fast access. This approach is openly inspired from the keyboard shortcut mechanism, well-known to the workstations users. This menu, called Command & Control Cube (abbreviation : C3) is an interface entirely designed for 3D. Evolutive and light, it allows to rapidly issue commands to an application with the help of a graphical and intuitive cubic structure and manipulation through quick hand movements in set directions. A same paradigm enables a manipulation mode for novice user based on the graphical feedback and a mode for expert user with rapid "eye-off" hand movements in space. After a set of formal evaluations, which validated the C3's principle, the menu was extended to a hierachical and a contextual version. Thus it covers for virtual environments the whole main control techniques of the pop-up menus.

In the same context, virtual environments lack certain fundamental tools, commonly found on the workstations in the 2D world. Thus, there is currently very few methods to enter text or numerical values inside a virtual environment. Mainly, the immersive nature of virtual environment configurations ban the use of the traditional keyboard and most of alternate cumbersome similar equipments. Therefore a method was developped to provide users with a simple and light typing method. It is adapted from a technique for PDA (Personal Digital Assistant) called Quikwrite.

Lastly, a multifunction tool for the exploration of 3D scenes has been developed. A virtual palette hold in the hand is composed of a rectangular surface that can be used interactively by the user to observe the surrounding scene through filters. The filters yield all sort of operations : mirror effect, zoom, altered lighting, cut of models, etc. A particular function of the reflecting mode help finding back objects that has been lost out of the field of view, which can happen on semi-immersive configurations like the workbench which displays only one half of the space.

# Table des matières

<b>Remerciements</b>	i
<b>Résumé</b>	i
<b>Abstract</b>	ii
<b>Table des matières</b>	iv
<b>Table des figures</b>	x
<b>1 Introduction</b>	1
1.1 La réalité virtuelle . . . . .	1
1.1.1 Définition . . . . .	1
1.1.2 Les facteurs d'immersion . . . . .	3
1.1.2.1 La vision . . . . .	4
1.1.2.2 Le toucher . . . . .	5
1.1.2.3 Autres sens . . . . .	7
1.2 Les configurations de réalité virtuelle . . . . .	8
1.2.1 Définition . . . . .	8
1.2.2 Principales configurations . . . . .	8
1.2.2.1 Le visio-casque . . . . .	8
1.2.2.2 Les environnements virtuels à base de projection sur grands écrans . . . . .	9
1.2.2.2.1 Les murs de projection . . . . .	9
1.2.2.2.2 VisionStation <sup>TM</sup> . . . . .	9
1.2.2.2.3 La CAVE <sup>TM</sup> . . . . .	11
1.2.2.2.4 Le plan de travail virtuel . . . . .	11
1.2.2.3 Equipements pour l'interaction . . . . .	12

1.3	Un nouvel enjeu . . . . .	13
<b>2</b>	<b>Etat de l'art</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.2	Equipements physiques de l'interaction . . . . .	19
2.2.1	Classification . . . . .	19
2.2.1.1	Equipements tenus en main . . . . .	20
2.2.1.2	Equipements portés . . . . .	21
2.2.2	Equipements génériques d'interaction en réalité virtuelle . . . . .	22
2.2.2.1	Equipements tenus en main . . . . .	22
2.2.2.2	Equipements portés . . . . .	26
2.2.3	Conclusion . . . . .	27
2.3	Navigation . . . . .	27
2.3.1	Déplacement par des moyens réels . . . . .	28
2.3.1.1	Marche . . . . .	28
2.3.1.2	Simulation de marche . . . . .	28
2.3.1.2.1	La marche sur place . . . . .	29
2.3.1.2.2	Tapis roulants . . . . .	29
2.3.1.2.3	Tapis munis de capteurs . . . . .	29
2.3.1.2.4	Plate-forme pédestre programmable . . . . .	31
2.3.1.3	Poste de conduite . . . . .	31
2.3.2	Déplacement par des moyens virtuels . . . . .	34
2.3.2.1	Conduite . . . . .	34
2.3.2.2	Sélection de destination . . . . .	34
2.3.2.3	Manipulation de la caméra . . . . .	35
2.3.2.3.1	Manipulation directe . . . . .	35
2.3.2.3.2	Manipulation indirecte . . . . .	35
2.3.3	Conclusion . . . . .	36
2.4	Sélection et Manipulation . . . . .	36
2.4.1	Sélection-manipulation avec la main . . . . .	38
2.4.1.1	Main virtuelle . . . . .	38
2.4.1.2	Go-Go . . . . .	39
2.4.1.3	Stretch Go-Go . . . . .	39
2.4.1.4	Travail bi-manuel . . . . .	40
2.4.2	Sélection-manipulation avec un pointeur . . . . .	42

2.4.2.1	Sphère de sélection . . . . .	42
2.4.2.2	Lancer de rayon . . . . .	42
2.4.2.3	Cône de sélection . . . . .	43
2.4.2.4	Diaphragme . . . . .	43
2.4.3	Plan image . . . . .	44
2.4.4	Sélection-manipulation à différentes échelles . . . . .	46
2.4.4.1	Changement d'échelle . . . . .	46
2.4.4.2	Monde en miniature . . . . .	47
2.4.5	Techniques de manipulation hybrides . . . . .	47
2.4.6	Techniques de manipulation multi-modales . . . . .	48
2.4.7	Conclusion . . . . .	49
2.5	Contrôle d'application . . . . .	50
2.5.1	Sélection verbale . . . . .	52
2.5.2	Sélection gestuelle . . . . .	53
2.5.3	Sélection d'outils virtuels . . . . .	53
2.5.4	Sélection d'outils physiques . . . . .	54
2.5.5	Interfaces matérielles dédiées au contrôle . . . . .	54
2.5.6	Sélection graphique . . . . .	55
2.5.6.1	Menus 2D implantés dans le monde 3D . . . . .	56
2.5.6.1.1	Positionnement : libre dans le monde . . . . .	57
2.5.6.1.2	Positionnement : centré sur un équipement mobile	57
2.5.6.1.3	Positionnement : centré sur un équipement fixe .	58
2.5.6.1.4	Positionnement : centré sur le corps . . . . .	58
2.5.6.2	Menus 3D . . . . .	60
2.5.6.2.1	Paradigmes 2D adaptés ou étendus au monde 3D	60
2.5.6.2.2	Paradigmes 3D . . . . .	62
2.5.6.3	Evaluations . . . . .	64
2.5.6.4	Conclusion . . . . .	64
2.5.7	Saisie de chaînes de caractères . . . . .	64
2.5.8	Conclusion . . . . .	68
2.6	Conclusion . . . . .	69
<b>3</b>	<b>Contrôle d'application en environnement virtuel : le <math>C^3</math></b>	<b>71</b>
3.1	Introduction . . . . .	71
3.2	Objectifs . . . . .	72

3.3	Les <i>Marking Menus</i> . . . . .	74
3.4	Le <i>Command and Control Cube</i> . . . . .	76
3.4.1	Division de l'espace sous forme cubique . . . . .	76
3.4.2	Le dispositif physique de commande . . . . .	77
3.4.3	Principe de fonctionnement . . . . .	79
3.4.4	Considérations de visibilité . . . . .	80
3.4.5	Modes novice et expert, un seul paradigme . . . . .	83
3.4.6	Fondements du mode aveugle . . . . .	85
3.5	Evaluations expérimentales et analyses . . . . .	86
3.5.1	La version du <i>C<sup>3</sup></i> . . . . .	88
3.5.2	Les Sujets . . . . .	88
3.5.3	La Tâche . . . . .	88
3.5.4	Données Collectées . . . . .	90
3.5.5	Résultats et discussions . . . . .	91
3.5.6	Premières conclusions . . . . .	94
3.6	Version hiérarchique . . . . .	94
3.7	Version contextuelle . . . . .	97
3.8	Info bulles . . . . .	99
3.9	Conclusion . . . . .	99
<b>4</b>	<b>Saisie de chaînes de caractères</b>	<b>103</b>
4.1	Introduction . . . . .	103
4.2	Le clavier virtuel . . . . .	104
4.2.1	Technique d'interaction . . . . .	105
4.2.2	Placement . . . . .	105
4.2.3	Evaluations informelles . . . . .	106
4.2.4	Conclusion . . . . .	107
4.3	QuikWrite VR . . . . .	107
4.3.1	Introduction . . . . .	107
4.3.2	QuikWrite . . . . .	108
4.3.3	Principe d'écriture . . . . .	110
4.3.4	Quikwrite VR . . . . .	113
4.3.4.1	Écriture au stylus . . . . .	113
4.3.4.2	Résolution des mouvements diagonaux . . . . .	114
4.3.4.3	Lisibilité et flexibilité de l'affichage . . . . .	115

4.3.4.4	Evaluation informelle . . . . .	117
4.4	Conclusion et perspectives . . . . .	118
<b>5</b>	<b>Exploration de scènes 3D à l'aide de la palette virtuelle</b>	<b>119</b>
5.1	Introduction . . . . .	119
5.2	Exploration de scènes 3D et utilisation des filtres . . . . .	120
5.2.1	Intégration des filtres en 3D . . . . .	121
5.2.2	Techniques de contrôle des filtres 3D . . . . .	122
5.3	Solutions proposées . . . . .	124
5.3.1	Introduction . . . . .	124
5.3.2	La palette virtuelle . . . . .	125
5.3.3	Le <i>Magic Mirror</i> . . . . .	127
5.3.3.1	Le miroir comme outil d'exploration . . . . .	127
5.3.3.2	Version de base du <i>Magic Mirror</i> . . . . .	128
5.3.3.3	Extensions du <i>Magic Mirror</i> . . . . .	129
5.3.3.3.1	Taille . . . . .	129
5.3.3.3.2	Ouverture . . . . .	129
5.3.3.3.3	Grossissement . . . . .	130
5.3.3.3.4	Inversion de l'image . . . . .	130
5.3.3.3.5	Dégagement de la vue du miroir . . . . .	130
5.3.3.3.6	Basculement vers la vue du miroir . . . . .	131
5.3.3.3.7	Application de filtres . . . . .	131
5.3.3.4	Avantages pour l'exploration . . . . .	132
5.3.3.4.1	Exploration multi-échelle . . . . .	132
5.3.3.4.2	Exploration d'un objet sous de multiples angles .	133
5.3.3.4.3	Explorer à travers une surface opaque . . . . .	133
5.3.3.4.4	Explorer l'intérieur de petits objets . . . . .	134
5.3.3.4.5	Retrouver des objets hors du champ de vision .	134
5.3.3.5	Réalisation et détails techniques . . . . .	135
5.3.4	Magic Windows . . . . .	137
5.3.5	Evaluations informelles . . . . .	138
5.4	Conclusions et perspectives . . . . .	139
<b>6</b>	<b>Minifly : une plate-forme logicielle pour environnement virtuel</b>	<b>141</b>
6.1	Présentation . . . . .	141

6.2	Affichage . . . . .	142
6.3	Gestion des outils d'interaction . . . . .	143
6.4	Applications . . . . .	145
<b>7</b>	<b>Conclusion et perspectives</b>	<b>147</b>
7.1	Conclusion . . . . .	147
7.2	Travaux futurs et perspectives . . . . .	154

# Table des figures

1.1	Bras à retour d'effort <i>PHANToM<sup>TM</sup></i> , copyright Sensable technologies . . . . .	6
1.2	Bras exosquelette Cyberforce, copyright Virtual Technologies . . . . .	7
1.3	Visio-casque, modèle 5DT HMD 800-35 de la compagnie 5DT <sup>TM</sup> . . . . .	9
1.4	Un mur de projection, copyright Barco . . . . .	10
1.5	VisionStation <sup>TM</sup> , copyright Elumens . . . . .	10
1.6	Système CAVE <sup>TM</sup> à 4 écrans, copyright université de l'Illinois . . . . .	11
1.7	Plan de travail virtuel à deux écrans, copyright i3d INRIA . . . . .	12
1.8	<i>Cyberglove</i> ou gants de données . . . . .	13
1.9	Stylus, copyright Polhemus . . . . .	14
1.10	Palette . . . . .	14
1.11	Lunettes actives CrystalEyes <sup>TM</sup> . . . . .	15
2.1	Taxonomie des équipements physiques de l'interaction . . . . .	20
2.2	Une <i>bat</i> équipée de plusieurs boutons, par Shaw et Green [107] . . . . .	23
2.3	Un <i>Wand</i> . . . . .	24
2.4	Utilisation d'une tête de poupée et d'un plateau transparent pour simuler les mouvements d'une tête virtuelle et d'un plan de coupe . . . . .	25
2.5	La <i>Cubic Mouse</i> de Frohlich et al. [46] . . . . .	26
2.6	La table d'interaction de Hachet et Guitton [58] . . . . .	26
2.7	Taxonomie des méthodes de navigation . . . . .	28
2.8	Le système GAITER (Templeman, Naval Research Labs) . . . . .	29
2.9	Tapis unidirectionnel, copyright Sarcos . . . . .	30
2.10	Tapis omni-directionnel, copyright Iwata 1996 . . . . .	30
2.11	Le <i>Virtual Motion Controller</i> , un tapis muni de capteurs, copyright Human Interface Technology Lab, Washington . . . . .	31
2.12	Tapis à capteurs de poids . . . . .	32
2.13	Magic Carpet, Tapis à capteurs piezo-électriques . . . . .	32

2.14	Le système Gaitmaster 1 d'Iwata . . . . .	33
2.15	Déplacement utilisant une bicyclette fixe (Sarcos Uniport [103]) . . . . .	33
2.16	Taxonomie des méthodes de sélection/manipulation . . . . .	37
2.17	Translation et rotation à deux mains, copyright Cutler et al. 1997 . . . . .	41
2.18	Opération de grossissement à deux mains, copyright Cutler et al. 1997 . . .	41
2.19	Le cône de sélection dans le système de modélisation JDCAD . . . . .	44
2.20	La technique <i>Sticky Finger</i> pour attraper un objet virtuel distant . . . . .	45
2.21	La technique "Sticky Finger" élargie par un cercle de sélection de rayon variable . . . . .	45
2.22	Principe du <i>Headcrusher</i> . . . . .	45
2.23	La technique <i>Headcrusher</i> pour attraper un objet virtuel distant . . . . .	46
2.24	Monde en miniature . . . . .	47
2.25	Taxonomie des méthodes de contrôle d'application . . . . .	51
2.26	Le <i>virtual tricorder</i> contrôle la position des fenêtres et permet de naviguer parmi les options. . . . .	58
2.27	Utilisation d'un PDA pour contrôler les applications au sein d'un environnement virtuel immersif. . . . .	59
2.28	Menu en anneau dans le système de modélisation JDCAD [80] . . . . .	60
2.29	Menu en demi-anneau avec pointeur virtuel de Wesche [122] . . . . .	61
2.30	Menu radial dans le système Holosketch [35] . . . . .	61
2.31	Le sundial menu de Chris Shaw et Mark Green . . . . .	62
2.32	Menu TULIP de Doug Bowman . . . . .	63
2.33	Le système de saisie de texte <i>VType</i> . . . . .	66
2.34	Saisie de texte à l'aide de <i>pinch gloves</i> par Bowman . . . . .	67
2.35	Le système de saisie de texte FingeRing . . . . .	67
2.36	Le système de saisie de texte VType . . . . .	68
3.1	Sélection dans un marking menu en mode novice avec le retour visuel (a) ou en mode expert en aveugle (b) . . . . .	75
3.2	Utilisateur appelant le Command and Control Cube ( $C^3$ ) à l'intérieur d'une application . . . . .	76
3.3	Espace divisé en 3x3x3 petits cubes . . . . .	77
3.4	Notre dispositif : des boutons de souris montés sur bagues et un capteur de position . . . . .	78
3.5	Les mouvements de la main sont reproduits sur la sphère-pointeur . . . . .	79

3.6	Le pointeur sphère est arrêté par l'enveloppe du $C^3$ . . . . .	80
3.7	Une icône est placée en texture sur le sommet des petits cubes . . . . .	80
3.8	Les trois étages du $C^3$ affichés ensemble posent des problèmes de visibilité des cubes et de leurs icônes . . . . .	81
3.9	Un seul étage du $C^3$ est affiché à un instant donné, selon la hauteur de la main : basse, médiane, haute . . . . .	81
3.10	Le $C^3$ avec son enveloppe transparente . . . . .	83
3.11	Affichage du $C^3$ selon la direction de regard . . . . .	84
3.12	Bip sonore au passage d'une frontière entre cases . . . . .	87
3.13	Vibration tactile au passage d'une frontière entre cases . . . . .	87
3.14	Cases à demi "remplies" . . . . .	88
3.15	La cible désignant la case à sélectionner à chaque étape du test . . . . .	89
3.16	Le dispositif de test . . . . .	90
3.17	Nombre de réponses correctes en fonction de la position des cases (nombre vert : 100% correct ; nombre bleu : score au dessus du premier quartile ; nombre rouge : score en dessous du premier quartile ; nombre noir : scores restants) . . . . .	93
3.18	Numérotation des cases . . . . .	93
3.19	Résultats en mode visuel et aveugle (cases vertes : score au dessus du premier quartile, cases rouges : score en dessous du premier quartile, cases grises : scores intermédiaires) . . . . .	93
3.20	Le $C^3$ hiérarchique . . . . .	96
3.21	Descentes successives dans deux sous-menus du $C^3$ hiérarchique . . . . .	97
3.22	Utilisateur appelant le Command and Control Cube ( $C^3$ ) à l'intérieur d'une application . . . . .	98
4.1	Un PDA (Palm V <sup>TM</sup> ), copyright Palm inc. . . . .	108
4.2	Les 9 zones de QuikWrite . . . . .	109
4.3	Un PDA (Palm V <sup>TM</sup> ) avec un masque pour QuikWrite, copyright Palm inc. . . . .	109
4.4	Trajet du stylo sur la surface de sélection de QuikWrite pour la lettre A. . . . .	110
4.5	Trajet du stylo sur la surface de sélection de QuikWrite pour la lettres S. . . . .	111
4.6	Trajet du stylo sur la surface de sélection de QuikWrite pour la lettres K. . . . .	111
4.7	Les 9 zones de QuikWrite donnent accès à 4 modes : minuscules, majuscules, numérique et ponctuation . . . . .	112

4.8	QuikWrite VR, le système de saisie de chaînes de caractères en environnement virtuel . . . . .	114
4.9	Division réelle de la surface de travail de notre système QuikWrite . . . . .	115
5.1	Utilisation d'une fenêtre filtrante sur les données de la scène. . . . .	123
5.2	Le <i>Virtual Remote Control</i> sur la palette virtuelle de Coquillart et Wesche [31] . . . . .	125
5.3	Le pad de Schmalstieg et al. [104] . . . . .	126
5.4	Le <i>Magic Mirror</i> . . . . .	129
5.5	Frustum de dégagement du miroir et frustum de vue de la scène . . . . .	131
5.6	Calcul de la vue du miroir . . . . .	135
5.7	Le frustum de vue est représenté en traits bleus. Le plan bleu clair forme le bord droit du frustum. Lorsqu'il est activé, seul ce qui se situe à sa droite est dessiné. Le même procédé est suivi pour les plans gauche, haut et bas, avant et arrière. . . . .	136

# Chapitre 1

## Introduction

### 1.1 La réalité virtuelle

#### 1.1.1 Définition

La réalité virtuelle est un domaine de recherche jeune et en pleine expansion. Le terme est souvent appliqué dans différents contextes allant de la synthèse d'images à des configurations de réalité virtuelle interactive. Il n'existe pas une définition unique de la réalité virtuelle, la communauté des chercheurs étant encore partagée sur la meilleure façon de la définir. Les définitions, bien que conceptuellement proches, bornent un domaine dont l'étendue dépend souvent de la spécialité de celui qui la définit. Nous allons ici présenter ces différentes définitions et tenter de proposer une approche fédératrice.

A une extrémité du spectre de définition de la réalité virtuelle se trouve la synthèse d'image. Celle-ci a pour objectif la création d'images d'objets 3D dont le réalisme visuel donne à croire à leur existence. En cela elle met en scène une réalité, qui reste cependant toute virtuelle. C'est la définition la plus large de la réalité virtuelle, qui apparaît à toute forme de création informatique artificielle une illusion de réalité qualifiée de réalité virtuelle.

A l'autre extrémité se trouvent les configurations dites de réalité virtuelle. Ces configurations permettent à un ou plusieurs utilisateurs de s'immerger totalement ou partiellement

---

ment dans un univers 3D de synthèse à travers des stimuli mono- ou plurisensoriels. Une configuration typique de réalité virtuelle propose une vision stéréoscopique de scènes 3D générées par ordinateur, un suivi des mouvements de la tête et divers moyens d'interaction directe avec les éléments de la scène. Ces systèmes cherchent également à proposer des périphériques pour stimuler d'autres sens que la vue, comme des systèmes à retour d'effort qui simulent des contacts physiques avec les objets virtuels.

Le passage de la synthèse d'image à la réalité virtuelle ne doit pas être vu comme un saut mais comme un continuum progressif dépendant d'un facteur commun qui est l'immersion. Ce qui permet de qualifier une image de synthèse de réalité virtuelle est le sentiment d'immersion que peut ressentir son spectateur et le degré de crédibilité qu'il lui accorde. La qualité visuelle, les couleurs, l'éclairage, l'animation sont de multiples facteurs aidant une image générée par ordinateur à posséder une crédibilité, alors que la scène 3D représentée n'est qu'une image 2D affichée sur un moniteur.

Si l'immersion est difficile devant une image de moniteur, elle s'enrichit rapidement lorsque des stimuli supplémentaires et cohérents lui sont ajoutés. Les premiers ajouts des configurations de réalité virtuelle portent sur la vue. La plupart de ces configurations proposent une vision stéréoscopique. D'autres canaux sensoriels peuvent aussi être stimulés et des solutions apparaissent qui gèrent le retour d'effort, le sens du toucher, l'ouïe, ou même encore l'odorat. En permettant à l'utilisateur d'explorer son environnement et en simulant de manière crédible le comportement des retours sensoriels dans cet environnement virtuel, ces configurations augmentent considérablement l'impression d'immersion. La possibilité d'interagir avec l'environnement est un facteur primordial de la réalité virtuelle.

Il est important de noter qu'un comportement simulé ne doit pas forcément être réaliste mais se doit de répondre à une attente de l'utilisateur. Par exemple il est important de voir son doigt arrêté par un mur virtuel, à l'aide d'un dispositif à retour d'effort. Pourtant, un chirurgien examinant le corps d'un patient virtuel peut souhaiter voir des couches internes du cerveau, s'aidant de plans de coupe manipulables, ce qui dans la réalité n'est pas possible ! La force de la réalité virtuelle est parfois de pouvoir aller au delà de ce que peut proposer la réalité. Pour cette raison il est plus adapté de parler de crédibilité et non de réalisme comme caractéristique des environnements virtuels.

Selon les points de vue, la réalité virtuelle commence dès la synthèse d'images alors

que d'autres n'octroient ce titre que lorsque la possibilité d'interaction est ajoutée à l'environnement virtuel. Ces perceptions de la réalité virtuelle sont donc assez semblables ; le débat porte principalement sur la position de la frontière dans cette transition du graphique aux environnements virtuels.

Dans la suite, nous prendrons le parti de parler de réalité virtuelle lorsqu'une interface naturelle est proposée à l'utilisateur, lui permettant d'interagir librement et directement avec son environnement 3D, prolongeant et renforçant ainsi ce qu'on appelle le sentiment de présence. Ce sentiment est la conviction temporaire et illusoire d'être au sein d'un environnement réel ou possédant un certain degré de crédibilité. De nombreux facteurs peuvent contribuer à ce sentiment de présence [86, 108, 110, 116], également appelé immersion. Nous en détaillons un certain nombre ci-après.

### 1.1.2 Les facteurs d'immersion

L'impression d'immersion que nous ressentons face à un univers virtuel est la mesure de la crédibilité que nous accordons à un univers virtuel. Elle est liée à un très grand nombre de facteurs, potentiellement aussi riches que l'univers réel qu'elle imite ou étend.

Un univers virtuel existe avant tout par la stimulation des sens de l'être humain. Cette stimulation peut être mesurée sur une double échelle : le niveau de **qualité** de la sensation virtuelle reproduite et sa **pertinence**, c'est à dire son adéquation aux attentes de l'utilisateur. Le niveau de qualité de la stimulation sensorielle comprend le degré de perfection d'une sensation, le confort de sa réalisation. La pertinence intègre principalement (mais pas uniquement) les facteurs d'interaction, soit la réaction de l'environnement aux actions de l'utilisateur. Nous allons examiner ces deux aspects au regard de deux sens communément stimulés à l'heure actuelle dans les environnements virtuels : la vue et le toucher, tout en notant que les perceptions humaines ne sont pas dissociées et que la cohérence de ces modalités sensorielles entre elles est également un aspect fondamental de l'immersion.

### 1.1.2.1 La vision

La vision est le sens le plus important de la perception humaine et par conséquent le premier sens stimulé dans un environnement virtuel. La perception humaine est complexe et englobe différents facteurs (voir à ce sujet le *Traité de la Réalité Virtuelle* de Fuchs et al. [47, 48]). Les images sur les deux rétines sont décomposées selon des critères de profondeur, forme, couleur et mouvement. La perception de la profondeur résulte de nombreux facteurs : indices mono et binoculaires, convergence et accommodation. Les configurations de réalité virtuelle exploitent une partie de ces caractéristiques pour créer l'illusion d'un univers en relief.

La qualité visuelle du rendu graphique des dispositifs de réalité virtuelle est un facteur important jouant sur la qualité de l'immersion visuelle : couleur, forme, texture, éclairage [109]. La largeur du champ de vision couvert est un autre point d'importance, lié au nombre et à la taille des écrans, et à leur disposition (devant les yeux, à distance, etc.).

Les résolutions graphiques des configurations de réalité virtuelle sont encore faibles proportionnellement aux surfaces des écrans mis en jeu. Elles sont limitées par une technologie différente des stations de travail, mais en rapide évolution, et par les ressources graphiques qu'elles requièrent pour fonctionner. La vision stéréoscopique nécessite en effet le calcul de deux images par écran, et plusieurs écrans sont souvent nécessaires pour couvrir un large champ de vision. La somme de ces contraintes impose une puissance graphique conséquente pour produire simultanément, en temps réel et à haute résolution un tel groupe d'images.

La vision stéréoscopique résulte de la convergence des yeux sur un même point d'intérêt, générant deux points de vue différents, et de la faculté du cerveau humain à recréer une image en relief à partir de ces deux points de vue. Pour créer artificiellement une vue stéréoscopique d'une scène virtuelle, il suffit de présenter devant chaque œil une image distincte, générée pour un point de vue correspondant au niveau de l'œil considéré dans l'espace virtuel, et observant la scène 3D. La correspondance entre la position de l'œil réel et du point de vue virtuel doit être la plus rigoureuse possible, au risque de perdre l'effet stéréoscopique ou de provoquer une fatigue oculaire et éventuellement un sentiment de malaise. L'écartement des yeux varie cependant d'un sujet à un autre, et il est délicat en pratique d'étalonner l'environnement virtuel à chaque passage d'un utilisateur à la

recherche de la valeur idéale. Aussi les configurations fonctionnent-elles souvent à partir d'une valeur moyenne fixe.

Si l'utilisateur est leurré et amené à croire à des objets en relief à une distance fictive [15], chacun de ses yeux continuent à converger et accomoder sur la surface des écrans affichant les deux images 2D. Cette divergence entre la position perçue des objets de la scène et la position réelle de l'accommodation peut créer des effets de flou discordants.

L'observation d'un point de vue fixe est assez peu immersive, puisqu'elle correspond à l'observation d'une image fixe comme une carte postale, ou une image de télévision (dont l'image n'est pas contrôlée par la position de l'utilisateur et donc non pertinente). Aussi, presque toutes les configurations de réalité virtuelle réagissent aux mouvements de la tête. La direction de regard peut ainsi être enregistrée à l'aide de périphériques physiques comme des capteurs, pour permettre de simuler les conséquences des mouvements de la tête. Ainsi un utilisateur peut-il "tourner" autour d'un objet virtuel pour le voir sous différents angles, l'ordinateur calculant les points de vue des images devant chaque œil pour qu'ils s'adaptent aux mouvements effectifs de la tête.

### 1.1.2.2 Le toucher

Après la vision, un des principaux facteurs d'immersion vient de la capacité qu'a l'utilisateur d'interagir avec la scène située devant lui. Le réflexe spontané d'un utilisateur novice confronté pour la première fois à une image stéréoscopique, c'est à dire une image possédant un véritable relief 3D, est en général de tenter de toucher du doigt ou d'attraper les objets virtuels pour vérifier leur existence, leur présence. La possibilité d'agir sur la scène par le biais de son corps et principalement ses mains et de constater une réaction cohérente, est un facteur fondamental d'immersion.

Une différence primordiale entre les stations de travail standards et les configurations de réalité virtuelle est que ces dernières proposent un environnement où l'espace de visualisation n'est pas séparé de l'espace de manipulation, mais se superpose à lui. L'interaction y est directe, parfois à deux mains.

La qualité et la pertinence des sensations associées au toucher dépendent d'un ensemble



FIG. 1.1 – Bras à retour d’effort *PHANToM<sup>TM</sup>*, copyright Sensable technologies

de stimuli réunis sous le terme “haptique”<sup>1</sup>, recouvrant le retour d’effort et le retour tactile [77].

Les images affichées devant les yeux d’un utilisateur doivent être pertinentes vis à vis de la vue, mais également des actions du corps de l’utilisateur. Celui-ci s’attend par exemple à voir et sentir son bras arrêté par un mur virtuel, comme le ferait un mur réel, ou simplement éprouver une résistance à déplacer ou soulever un poids.

Ces objets présentés dans l’univers virtuel n’ayant pas d’existence réelle, il est souhaitable que la configuration de réalité virtuelle prenne en charge la détection des collisions et la gestion des contacts avec les éléments de la scène. La cohérence des sensations peut être accrue par une gestion dynamique des contraintes et un modèle physique pour les éléments virtuels de la scène [102].

De nombreux dispositifs ou “interfaces haptiques” ont été développés pour simuler le retour d’efforts sur les mains et/ou les bras de l’utilisateur. Plus ou moins encombrants ils peuvent varier du simple bras à retour d’effort (voir figure 1.1) jusqu’au bras exosquelette (voir figure 1.2).

---

<sup>1</sup>Le terme “haptique” vient du grec *haptēsthai* qui signifie sentir par le toucher.



FIG. 1.2 – Bras exosquelette Cyberforce, copyright Virtual Technologies

De fait, les interfaces haptiques ne se limitent pas au simple retour d'effort, qui désigne l'exercice de forces sur les membres du corps et qui constraint les mouvements, mais comprennent également le retour tactile qui simule des excitations de la peau et englobe des modalités très variées comme la température, la texture ou le contact.

#### 1.1.2.3 Autres sens

La stimulation des autres sens de l'homme peut accroître l'immersion en environnement virtuel. L'ouïe est un sens particulièrement intéressant pour augmenter le réalisme, aider à la localisation spatiale, ou comme canal d'information additionnel, parallèle, ou enfin comme modalité de substitution. Les autres sens sont moins couramment exploités, même si certains travaux existent concernant l'odorat.

## 1.2 Les configurations de réalité virtuelle

### 1.2.1 Définition

Un environnement virtuel est un univers 3D généré par ordinateur. Différents canaux sensoriels peuvent être stimulés pour mieux simuler l'illusion de présence, communément appelée immersion.

Les configurations de réalité virtuelle sont des installations permettant à un ou plusieurs utilisateurs d'expérimenter un environnement virtuel. Ces environnements peuvent être immersifs si l'utilisateur est entièrement "plongé" ou "immergé" dans l'univers 3D. Ils sont parfois dits semi-immersifs dans le cas où l'univers 3D occupe une portion de l'espace réel de l'utilisateur, mais que celui-ci garde contact avec le monde réel : son corps, ses collègues, la pièce dans laquelle il se trouve.

### 1.2.2 Principales configurations

Le premier sens simulé est généralement la vue. Il existe différentes approches et technologies pour recréer l'illusion d'un paysage virtuel. Deux grandes familles se distinguent à l'heure actuelle : les visio-casques et les dispositifs à base de projection sur grands écrans.

#### 1.2.2.1 Le visio-casque

Très populaires aux premières heures de la réalité virtuelle, les visio-casques sont des équipements individuels permettant de voir l'univers virtuel par le biais d'écrans placés devant les yeux (voir figure 1.3). Les mouvements de la tête sont enregistrés pour adapter la vue de l'univers virtuel à la direction de regard. Les visio-casques présentent encore quelques faiblesses, comme la basse résolution des écrans ou un champ de vision étroit. La vision entière étant remplacée par une vue du monde virtuel, l'utilisateur ne peut plus voir directement son corps. Cette situation peut être déconcertante, et il est souvent nécessaire de recréer sous forme d'avatars les bras et les mains de l'utilisateur dans l'espace virtuel, afin de permettre l'interaction manuelle.



FIG. 1.3 – Visio-casque, modèle 5DT HMD 800-35 de la compagnie 5DT<sup>TM</sup>

### 1.2.2.2 Les environnements virtuels à base de projection sur grands écrans

Les technologies à base de grands écrans les plus communes sont les murs de projection, le plan de travail virtuel et la CAVE<sup>TM</sup>. Ces configurations se distinguent par le nombre et la disposition des écrans. Nous détaillerons plus spécifiquement le Plan de Travail Virtuel, sur lequel les travaux de la thèse ont été effectués.

**1.2.2.2.1 Les murs de projection** La configuration la plus simple consiste en un unique écran stéréoscopique disposé verticalement. Plusieurs configurations de taille et de forme différentes existent. Un simple écran rectangulaire placé contre un mur est appelé communément un *Wall Display* ou mur de projection (voir figure 1.4) et agit comme une fenêtre sur le monde virtuel. Plus complexes, certaines installations comme les Reality Center<sup>TM</sup> de SGI<sup>TM</sup> s'apparentent à des salles de projection pour une large audience. Un large écran vertical cylindrique permet d'offrir aux spectateurs un très large champ de vision. elles emploient plusieurs projecteurs affichant bord à bord une image panoramique sur l'écran géant.

**1.2.2.2.2 VisionStation<sup>TM</sup>** Le constructeur Elumens propose une configuration de réalité virtuelle mono-utilisateur appelée VisionStation<sup>TM</sup>. L'utilisateur est assis en face d'un écran de travail demi-sphérique, et dispose d'un support à hauteur de mains pour poser clavier et souris (voir figure 1.5). Cette configuration originale permet de diviser l'environnement en deux : l'espace réel en arrière, l'espace virtuel devant soi.

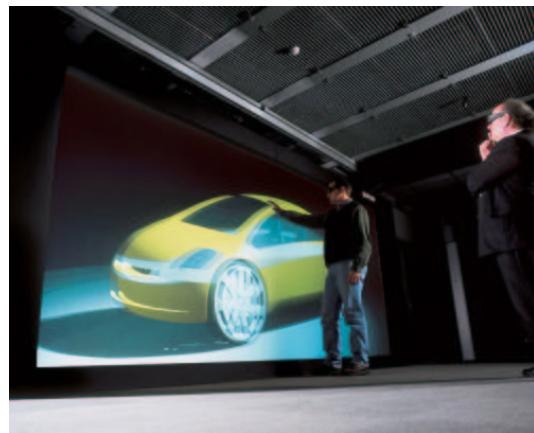


FIG. 1.4 – Un mur de projection, copyright Barco



FIG. 1.5 – VisionStation<sup>TM</sup>, copyright Elumens

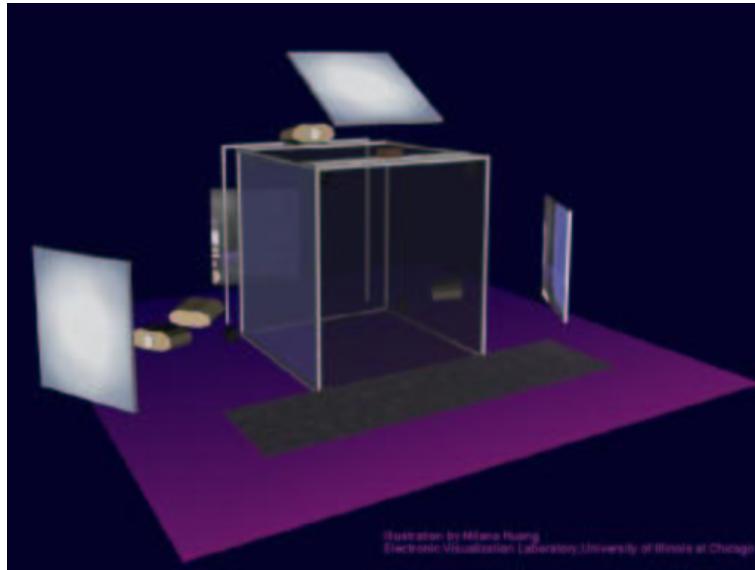


FIG. 1.6 – Système CAVE<sup>TM</sup> à 4 écrans, copyright université de l'Illinois

**1.2.2.2.3 La CAVE<sup>TM</sup>** Les CAVEs<sup>TM</sup> [32] sont des configurations d'environnements virtuels volumineuses composées de 4 à 6 écrans géants (environ 3 mètres de côté) disposés de manière à former les faces d'une pièce cubique (voir figure 1.6). Les utilisateurs entrent dans l'espace formé par ces murs pour interagir avec l'univers virtuel. Les modèles les plus simples à 4 écrans affichent des images stéréoscopiques sur 3 murs et le sol. Les modèles à 6 écrans utilisent un panneau coulissant pour laisser entrer les utilisateurs et "refermer" l'univers virtuel après leur passage. Les images sont rétroprojetées sur les écrans, à l'exception de l'écran du sol pour lequel l'image est projetée du plafond lorsque cela est possible, pour éviter la très complexe et coûteuse gestion d'un plancher-écran translucide et résistant, ainsi qu'une distance sous plancher de 2 ou 3 mètres pour le recul du projecteur. Les CAVEs<sup>TM</sup> sont des configurations fortement immersives, particulièrement adaptées à des tâches de navigation dans de grands espaces.

**1.2.2.2.4 Le plan de travail virtuel** Le *Plan de Travail Virtuel* [72, 71] (ou *Responsive Workbench* en anglais) est une configuration semi-immersive adaptée pour le travail de maquettes virtuelles, par exemple en bureau d'études. Il se présente comme une table dont la face supérieure est un écran, affichant des images stéréoscopiques.

Notre configuration de Plan de Travail Virtuel est constituée de deux écrans, permettant d'augmenter considérablement le champ de vision de l'utilisateur (voir figure 1.7). Le

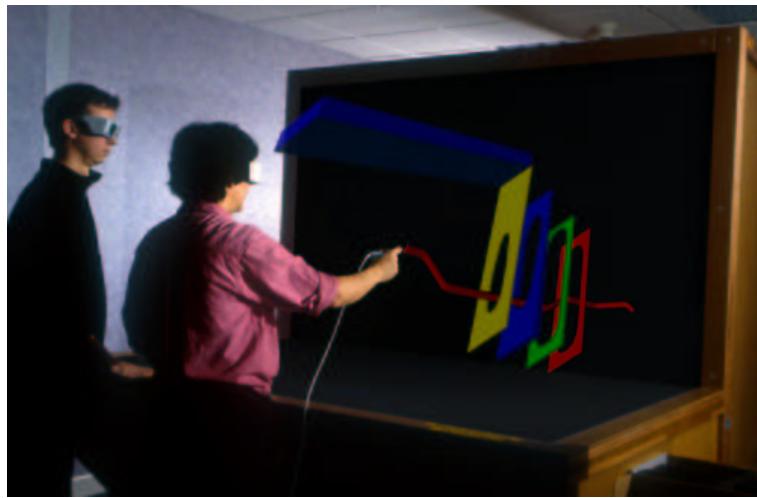


FIG. 1.7 – Plan de travail virtuel à deux écrans, copyright i3d INRIA

second écran est placé verticalement bord à bord avec l'écran horizontal. L'espace virtuel est compris dans une pyramide centrée sur la tête de l'utilisateur et s'appuyant sur les bords des deux écrans.

L'image est retro-projetée sur chaque surface translucide à l'aide d'un projecteur CRT tri-tubes (Rouge, Vert, Bleu) haute résolution, haute fréquence. Le projecteur est couplé à un miroir incliné à 45 degrés pour limiter l'encombrement spatial que nécessite le recul du projecteur sous la table.

L'image affichée est stéréoscopique, c'est à dire que l'écran affiche successivement et à très haute fréquence deux images différentes : une pour l'œil droit, une pour l'œil gauche de l'utilisateur. Des lunettes à cristaux liquides dites actives viennent compléter l'équipement pour filtrer ces images vers chaque œil, en obturant en synchronisme avec l'affichage la vue d'un œil ou de l'autre. La synchronisation s'effectue à l'aide d'un signal lumineux infra-rouge qu'un équipement couplé à l'ordinateur émet et que les lunettes reçoivent.

### 1.2.2.3 Equipements pour l'interaction

L'interaction entre l'utilisateur et l'application dans ces environnements virtuels immersifs est réalisée à l'aide de périphériques de manipulation 3D. Ce sont des équipements dotés de capteurs de position et d'orientation (comme des capteurs électro-magnétiques). Ils peuvent être portés comme par exemple des gants, ou tenus en main dans le cas d'ou-

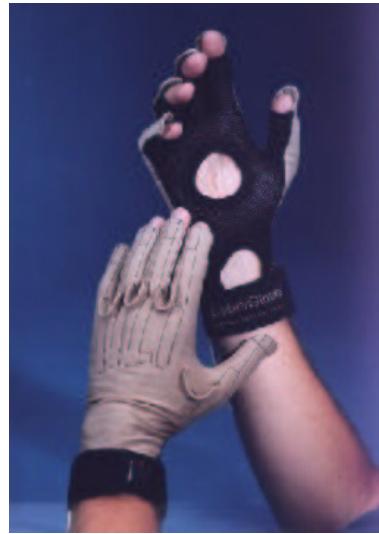


FIG. 1.8 – *Cyberglove* ou gants de données

tils :

- Gants de données [130] : gants équipés d'un capteur de position et dont la flexion des doigts est mesurée et enregistrée (voir figure 1.8)
- Stylus : stylo équipé d'un capteur de position (voir figure 1.9).
- Palette [31] : petite raquette composée d'une surface transparente et d'un manche, équipée d'un capteur de position (voir figure 1.10).
- Autres outils : beaucoup d'autres périphériques peuvent être construits à partir d'un objet simple équipé d'un capteur de position.
- Lunettes : Les configurations à base de grands écrans utilisent principalement des lunettes actives (obturation synchronisée des verres gauches et droits) ou passives (polarisation des verres) pour séparer la vue gauche de la vue droite, et observer ainsi une scène stéréoscopique. Elles sont souvent équipées d'un capteur de position pour permettre à l'utilisateur de modifier interactivement son point de vue sur la scène (voir figure 1.11).

### 1.3 Un nouvel enjeu

Les configurations d'environnement virtuel ont l'ambition de poursuivre et dépasser les objectifs de visualisation et de manipulation de données des applications équivalentes sur



FIG. 1.9 – Stylus, copyright Polhemus



FIG. 1.10 – Palette



FIG. 1.11 – Lunettes actives CrystalEyes<sup>TM</sup>

stations de travail de type écran, clavier, souris. Ils promettent une expérience plus riche, plus complète et une interaction plus naturelle et spontanée. L'utilisateur n'est plus assis en face d'une station de travail, communiquant à l'aide d'une interface homme-machine, il est lui-même immergé dans un univers virtuel à trois dimensions et agit directement sur les éléments de son application. La machine disparaît pour ne laisser que l'homme face à l'application qu'il utilise et le défi consiste alors à favoriser la communication homme-application en proposant des interfaces transparentes et intuitives.

Dans un domaine encore jeune, l'absence d'interfaces répandues, populaires et efficaces, de métaphores communes pour l'interaction, sont les obstacles à franchir pour tirer pleinement parti de ces configurations aux contraintes et aux modalités nouvelles.

Dans **le second chapitre** de ce mémoire, nous présentons l'état de l'art des différentes techniques d'interaction en environnement virtuel à travers différentes taxonomies, avec un regard particulier sur les techniques de sélection/manipulation et le contrôle d'application qui sous-tendent les travaux de la thèse.

**Le troisième chapitre** propose un nouveau paradigme de contrôle d'application pour les environnements virtuels 3D immersifs. Appelé *Command and Control Cube* (abrégé en *C<sup>3</sup>*), il se caractérise par une interface graphique discrète combinée à une interaction gestuelle simple et rapide. Il propose à travers un paradigme commun deux niveaux d'interaction, pour novice ou pour expert et s'adapte automatiquement au niveau d'ex-

périence de l'utilisateur. Des évaluations formelles poursuivent cette présentation, ainsi qu'une extension du concept du  $C^3$  à une version hiérarchique et contextuelle.

**Un quatrième chapitre** explore la saisie de texte dans les environnements virtuels. Après une première étude d'un clavier virtuel, une technique d'écriture légère baptisée *QuikWrite VR* est développée. Elle s'inspire d'une technique d'écriture sur PDA (*Personal Digital Assistant*) développée par Ken Perlin : *QuikWrite* [94]. A l'instar du stylet sur PDA, elle utilise le stylus (un stylo équipé d'un capteur) pour écrire sur une surface plane virtuelle s'approchant en apparence et en concept du Command and Control Cube.

L'exploration de scènes virtuelles 3D est l'objet **du cinquième chapitre**. La palette virtuelle de Coquillart et Wesche [31] sert de support à deux nouvelles métaphores pour l'exploration : le miroir et la loupe. A travers ces métaphores, c'est le concept plus général des filtres qui est intégré dans l'univers virtuel. Ces deux approches sont baptisées *Magic Mirror* et *Magic Windows*. Leur pouvoir d'exploration est étendu et augmenté par une série de propriétés additionnelles, impossibles dans le monde réel.

L'ensemble des outils et techniques d'interaction précédentes ont été intégrées dans une plate-forme logicielle de développement pour le Plan de Travail Virtuel : Minifly. Cette plate-forme a été mise au point dans le cadre de cette thèse pour répondre à un double objectif : offrir un support et un cadre commun, formel et robuste à la visualisation d'objets 3D et au développement d'outils d'interaction sur le Plan de Travail Virtuel. Minifly sera décrit dans **le sixième chapitre** de ce mémoire.

**Une conclusion** viendra synthétiser les résultats de la thèse, leurs réponses aux enjeux du contrôle d'application et de l'exploration virtuelle, et ouvrira des perspectives futures dans ces domaines de la réalité virtuelle.

# Chapitre 2

## Etat de l'art

### 2.1 Introduction

Les configurations d'environnement virtuel sont de nouveaux média pour la visualisation et l'exploration immersive et interactive de données 3D générées par ordinateur. Pour exploiter ces configurations matérielles, les applications doivent produire des solutions pour la génération de scènes 3D, la visualisation stéréoscopique et interactive de ces scènes, la gestion des paradigmes d'interaction et le contrôle d'application.

La génération de scènes 3D correspond au domaine déjà longuement exploré de la synthèse d'images. La visualisation dans les environnements virtuels doit prendre en compte une vision souvent stéréoscopique et associée au point de vue de l'utilisateur, suivant les mouvements et les inclinaisons de la tête. Des paradigmes d'interaction doivent être proposés pour permettre aux utilisateurs d'interagir avec l'univers virtuel. Contrairement aux environnements 2D de type moniteur, clavier et souris, l'utilisateur est immergé, totalement ou partiellement, dans l'univers virtuel. L'espace de visualisation coïncide avec l'espace de manipulation. La nature même des interactions en est bouleversée, aussi bien par l'ajout de la troisième dimension que par le caractère "direct" des interactions.

Dès lors que les anciens périphériques, clavier et souris, ne sont plus adaptés à un environnement devenu virtuel et immersif, de nouveaux outils d'interaction directe doivent être proposés pour les remplacer. Nous présenterons dans la première partie de cet état de

---

l'art une classification des différents outils d'interaction en environnement virtuel et appliquerons cette classification aux équipements que nous retrouverons dans les techniques d'interaction qui les emploient.

Pour faire face à ce changement d'environnement, mais également de moyens d'interagir sur cet environnement, les applications doivent s'adapter et reconcevoir leurs techniques et parfois leurs paradigmes d'interaction. La conception d'interfaces et de techniques de manipulation 3D ne peut simplement reposer sur les techniques précédentes du monde 2D, car elle suscite de nouvelles difficultés, qui nécessitent de comprendre et d'aborder les enjeux d'un univers immersif à trois dimensions [61, 73]. Diverses études ont cherché à regrouper et classifier ces nouvelles méthodes d'interaction pour organiser leur recherche et fournir de grandes lignes directrices aux concepteurs d'interface immersive 3D [59, 98, 19, 17].

Les différentes formes d'interaction possibles sont généralement regroupées à l'intérieur de trois grandes catégories ou tâches génériques [59, 38] : la *navigation* dans l'univers virtuel, la *sélection/manipulation* d'objets 3D, et le *contrôle d'application*. Les deux premières s'occupent des interactions entre l'utilisateur et les données de l'application. La dernière catégorie est fondamentale à toute application puisqu'elle s'occupe du dialogue entre l'utilisateur et l'application elle-même. Le terme interface homme-application sera d'ailleurs préféré ici à celui plus ancien d'interface homme-machine. L'interface est traditionnellement le médium de communication entre l'homme et la machine. Dans les environnements virtuels cependant, la "machine" n'est pas visible ; l'utilisateur est immergé et directement confronté à l'application. Il est d'autant plus impératif pour l'interface immersive de se faire oublier au profit d'un dialogue transparent et direct entre l'homme et l'application.

Nous détaillerons dans la suite chacune de ces trois grandes tâches génériques, en portant un regard plus approfondi sur le contrôle d'application, qui sous-tend les travaux de cette thèse. Dans notre état de l'art, nous nous concentrerons plus particulièrement sur les interfaces graphiques, abordant plus brièvement les interfaces gestuelles et à commandes vocales. Les aspects haptiques forment également un domaine d'étude à part entière, qui ne sera pas spécifiquement abordé ici, car il s'éloigne de nos travaux.

## 2.2 Equipements physiques de l'interaction

Devant un écran de station, l'utilisateur désigne, manipule et contrôle les objets graphiques à l'aide d'un dispositif de pointage, la souris. Les configurations de réalité virtuelle immergeant l'utilisateur dans un univers virtuel à trois dimensions. Pour agir dans cet environnement et dialoguer avec l'application, l'utilisateur doit être en mesure de maîtriser des données spatiales et engendrer des informations spatiales.

Les techniques de manipulation en environnement virtuel reposent sur des dispositifs matériels pour la détection des positions et des orientations. Il existe des solutions à base de caméras, dont les images sont analysées pour en extraire les informations spatiales. Elles sont encore prospectives et complexes à mettre en œuvre. A l'heure actuelle, la solution la plus répandue consiste à employer des capteurs électromagnétiques, servant à repérer des mouvements de translation et des orientations. Ils sont l'interface physique qui permet le dialogue entre l'utilisateur et l'application. Dans la suite nous parlerons simplement de "capteurs de position" ou même de "capteurs" pour désigner ces capteurs à 6 degrés de liberté, par souci de simplicité d'écriture. Si des capteurs simples peuvent être utilisés tels quels pour la manipulation, leur forme est peu maniable [63]. Leur apparence est donc souvent retravaillée pour faciliter l'interaction et la rendre plus naturel à l'homme. Les informations provenant des capteurs étant des signaux continus, ils sont souvent agrémentés de boutons, pour fournir des signaux d'entrée discrets.

### 2.2.1 Classification

Il existe une grande variété de formes, d'apparences et d'usages pour tous ces équipements. A notre connaissance, aucune méthode de classement n'a été proposée à ce jour. Afin de synthétiser les différentes approches, nous proposons ici une classification des équipements d'interaction en réalité virtuelle, selon que l'équipement soit porté, comme un vêtement ou une montre, ou tenu en main. Cette classification s'attache essentiellement à la phase d'utilisation elle-même des équipements et non leur état de repos (par exemple : posé sur le sol, ou suspendu). Le tableau de la figure 2.1 résume les différentes catégories de notre classement.

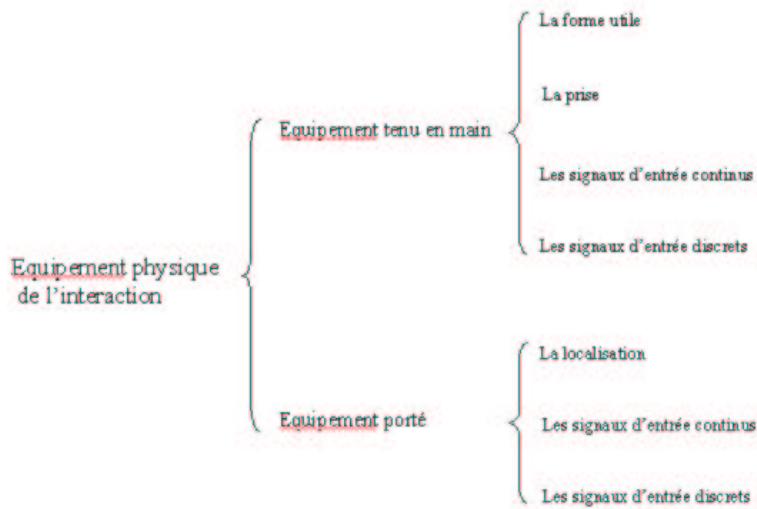


FIG. 2.1 – Taxonomie des équipements physiques de l'interaction

### 2.2.1.1 Equipements tenus en main

Dans le monde réel l'homme est amené à interagir avec de nombreuses machines aux fonctions variées et à réaliser de nombreuses tâches manuelles de diverses natures. Une technique ancestrale pour répondre à chaque défi de son environnement et faire face à cette diversité consiste à employer des outils. Cette approche est également très populaire en réalité virtuelle, puisque les exigences d'un monde virtuel sont très proches de celles du monde réel. Les équipements tenus en main s'apparentent à des outils que l'utilisateur emploie pour mener à bien des tâches qui peuvent être génériques ou spécialisées, selon la nature de l'outil et les techniques d'interaction qui peuvent lui être associées. Quatre caractéristiques primordiales peuvent être distinguées pour décrire et classifier les outils utilisés en réalité virtuelle :

- **La Prise** : Le premier contact d'un utilisateur avec un outil passe par la façon de le saisir. Cette prise est importante à plusieurs égards. Elle détermine tout d'abord le confort en manipulation de l'utilisateur. Un objet difficilement saisissable, ou nécessitant des tensions dans le bras, le poignet ou les doigts peut s'avérer très vite fatigant. La prise influe également sur l'aisance future de manipulation des objets virtuels. Puisque l'outil servira de médium pour désigner ou manipuler les objets virtuels, il doit être facile de lui donner diverses orientations. Enfin elle doit être de préférence familière, c'est à dire rappeler des interactions du monde réel. Une prise inhabituelle peut demander un temps d'apprentissage et/ou être moins spontanée

et donc moins efficace à long terme.

- **La Forme Utile** : Nous appelons forme utile tout ou partie de l'outil présentant une forme caractéristique utilisable comme support pour l'interaction. La forme peut servir entre autres à suggérer une fonction pour l'outil ou initier et matérialiser une forme géométrique qui sera complétée virtuellement, par exemple une droite ou un plan.
- **Les signaux d'entrée continus** : La plupart des équipements d'interaction manuelle en réalité virtuelle, sinon tous, comprennent un capteur de position et d'orientation dans l'espace. Un tel capteur permet de mesurer six degrés de liberté simultanément, trois en translation, trois en rotation. D'autres dispositifs peuvent également exister, agissant sur des degrés de liberté moins nombreux pour obtenir un contrôle plus réduit mais moins complexe selon un ou deux degrés de liberté, en translation ou en rotation.
- **Les signaux d'entrée discrets** : Les équipements d'interaction proposent couramment des boutons pour générer des événements discrets. Leur nombre peut varier d'un équipement à un autre. Ce sont en général des commutateurs à deux états, mais certains dispositifs peuvent exister possédant plus de positions.

### 2.2.1.2 Equipements portés

Agir par le biais d'outils tenus en main n'est pas la seule approche possible. Dans le monde réel nous portons souvent de petits appareils, gadgets, bijoux ou simplement vêtements, comme des montres, des bracelets ou des gants qui peuvent être autant d'idées pour équiper un utilisateur d'environnement virtuel d'outils d'interaction peu encombrants, peu fatiguants et libérant les mains. Ces accessoires, lorsqu'ils sont utilisés pour la réalité virtuelle peuvent se caractériser par les critères suivants :

- **La localisation** : Le premier élément de classement concerne l'endroit du corps où est accroché l'équipement (tête, cou, taille, bras, poignet, main, doigt, etc.). Il détermine son accessibilité dans le cadre d'un équipement devant être actionné par la main, par exemple, s'il présente des boutons. Le confort d'utilisation est directement relié à l'adéquation entre l'emplacement choisi sur le corps, et le poids et volume de l'équipement. Un dispositif léger et peu volumineux peut être plus avantageusement porté comme une bague, un autre plus lourd ou encombrant en bracelet au niveau

du poignet, voire accroché sur l'avant-bras. La mobilité physiologique de l'endroit choisi influe également sur l'aisance de manipulation, dans le cas d'un équipement servant à la désignation ou la manipulation d'objets virtuels.

- **Les signaux d'entrée continus** : L'utilisation d'un capteur de position et d'orientation est également courante pour les équipements portés, avec pour objectif cette fois de repérer le mouvement du corps. Ils sont parfois agrémentés de dispositifs complémentaires pour mesurer d'autres valeurs continues telle la flexion des doigts.
- **Les signaux d'entrée discrets** : Comme précédemment, les équipements portés peuvent présenter des boutons dont le nombre d'états et le nombre total peut varier selon les objectifs d'interaction poursuivis.

## 2.2.2 Equipements génériques d'interaction en réalité virtuelle

Nous présentons dans cette partie les équipements physiques courants de la réalité virtuelle, pouvant servir à de multiples tâches d'interaction. Certaines interfaces matérielles, dédiées à des tâches uniques, seront décrites dans les sections correspondantes de la navigation et du contrôle d'application.

### 2.2.2.1 Equipements tenus en main

Pour tous les équipements suivants tenus en main, nous retrouvons un capteur de position et d'orientation, que nous nommerons simplement capteur de position dans la suite par souci de commodité. Ce capteur permet de repérer l'outil dans l'espace ; il relie l'utilisateur au monde virtuel. C'est le principal signal d'entrée continu. L'évolution des périphériques est principalement liée à leur prise et la forme utile.

Le périphérique le plus simple que l'on puisse trouver, et un des premiers utilisés pour l'interaction 3D est la *bat* de Ware [24] (cf. figure 2.2), un simple capteur équipé d'un bouton. Une variante plus récente étend son concept avec un dispositif à double sensibilité en empilant deux boutons l'un sur l'autre. Une pression légère n'enfonce qu'un des deux boutons, tandis qu'une pression plus franche presse les deux. Cette technique est baptisée *pop-through* par Zeleznik et al. [126].

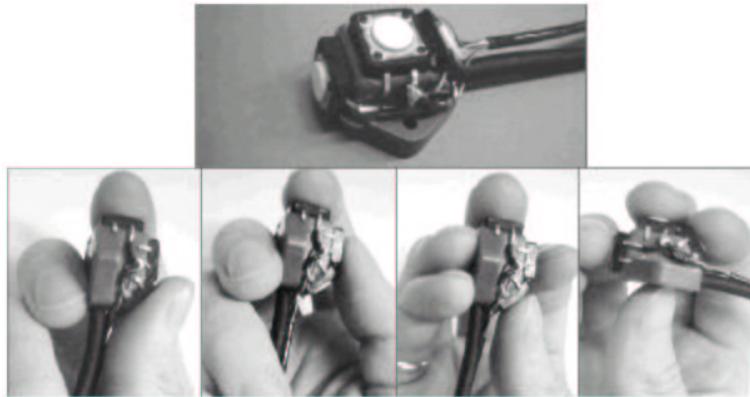


FIG. 2.2 – Une *bat* équipée de plusieurs boutons, par Shaw et Green [107]

Une des fonctionnalités recherchées pour ces périphériques 3D est de pouvoir remplir les tâches courantes d'une souris, dans un univers à trois dimensions, à savoir désigner avec précision des points ou des portions de l'espace, tout en fournissant un, deux ou trois signaux discrets d'entrée à l'application, c'est à dire les boutons. Une première approche, directe, a ainsi consisté à doter une souris 2D d'un capteur de position [124]. La souris étant conçue pour être posée et manipulée sur un plan, sa prise n'est pas complètement adéquate à un univers à trois dimensions, mais cette première approche est simple et fonctionnelle.

Suivant le même objectif, un bâton ou *wand* [105] en anglais (cf. figure 2.3) est une forme d'outil 3D plus naturelle, qui se retrouve dans de nombreux travaux d'interaction. Le *wand* se présente comme un manche cylindrique équipé de boutons, une forme simple et adaptée pour désigner ou fixer des orientations. Elle rappelle une lampe-torche et sa prise en main comme sa manipulation est spontanée et aisée.

*Le stylus* est une variante moderne de ce dernier périphérique, très adapté à la manipulation de maquettes numériques. Il se retrouve en conséquence couramment sur le plan de travail virtuel. Il se présente comme un gros stylo, rappelant le *wand* mais avec une extrémité effilée pour la pointe. Un ou plusieurs boutons sont généralement présents le long du manche. Il est intéressant de noter que spontanément, les utilisateurs l'utilisent indifféremment comme une baguette avec une prise à pleine main, refermant tous les doigts sur le manche, ou comme un stylo, en posant la tige entre le pouce et l'index. La prise sous forme de stylo, très familière, permet plus de liberté en rotation et est généralement plus efficace pour les tâches de manipulation. La versatilité de prise reste cependant un



FIG. 2.3 – Un *Wand*

atout du stylus, qui peut servir différents paradigmes d’interaction.

La forme utile du stylus et du *Wand*, un segment cylindrique, privilégie une direction de l'espace et dirige naturellement ces deux périphériques vers des techniques de manipulation de droites, ainsi que vers des techniques en tirant parti, comme la désignation d'objets par pointage. La pointe effilée du stylus incite de plus à des travaux de précision.

Partageant également un manche cylindrique comme prise, la palette [31] (voir figure 1.10) est un autre équipement populaire des configurations de réalité virtuelle. Elle se présente comme une petite raquette, composée d'un manche et d'une plaque transparente. L'ajout de la plaque comme surface utile permet d'utiliser la palette comme support à de nombreuses opérations nécessitant la matérialisation d'un plan : manipulation de plan de coupe, affichage de menus 2D, écriture de texte sur plan, etc.

Suivant une voie différente, un simple capteur peut être entouré cette fois d'une coque sphérique, permettant une prise en main confortable et facilitant les rotations par l'homogénéité de sa surface [63].

Pour des tâches moins génériques, des périphériques peuvent être construits à partir d'un capteur et d'une forme utile adéquate. Appelés *props*<sup>1</sup>, ils représentent généralement des objets virtuels de la scène. Ils contribuent à une manipulation plus naturelle et plus

---

<sup>1</sup>Le terme anglais *prop* n'est généralement pas traduit. Il désigne un outil physique qui sert de support à l'interaction.

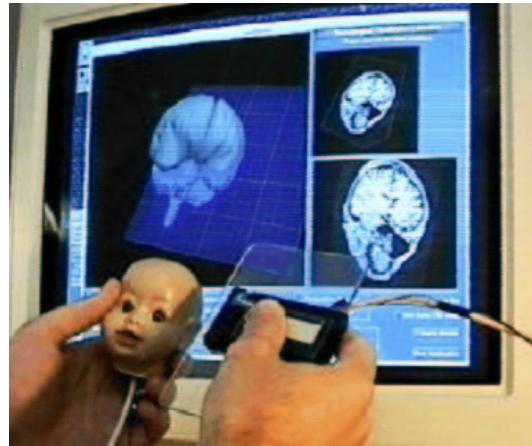


FIG. 2.4 – Utilisation d'une tête de poupée et d'un plateau transparent pour simuler les mouvements d'une tête virtuelle et d'un plan de coupe

efficace, grâce à un retour sensoriel tactile. Hinckley en donne l'exemple dans une application de chirurgie [62], où le médecin doit examiner sur un écran à l'aide de plans de coupe des données volumétriques représentant la tête de son patient. Une poupée et une plaque rectangulaire matérialisent respectivement la tête virtuelle du patient et le plan de coupe (voir figure 2.4). Tenus en mains, il devient alors très simple d'orienter le plan par rapport à la tête et d'explorer les données volumétriques.

La manipulation de multiples plans de coupe est l'objet d'une autre étude. Frohlich et Plate [46] présentent un nouveau périphérique nommé *Cubic Mouse* : une petite boîte cubique repérée dans l'espace par un capteur, équipée de boutons et traversée selon chacune de ses faces par des tiges mobiles (voir figure 2.6). Les trois tiges traversant le cube apportent des signaux continus additionnels par rapport aux équipements précédents. Leur forme utile privilégie trois orientations particulières de l'espace, sous forme de droites, et la possibilité de générer un déplacement sur ces axes fournit naturellement trois glissières mesurant des déplacements continus selon ces droites. Le concept général derrière la création de ce périphérique suggère d'utiliser le socle cubique pour matérialiser un objet ou un bloc de données 3D à examiner, et de déplacer les trois tiges à travers le cube pour bouger trois plans de coupe orthogonaux dans la scène.

Hachet et Guitton proposent un dernier périphérique intitulé *Interaction Table* [58]. Cette table d'interaction est composée d'un plateau en forme de disque plat, mobile selon les six degrés de liberté, au sommet d'un pilier. Le périphérique utilise à la fois des informations isotoniques pour les orientations de la table et isométriques pour ses trans-

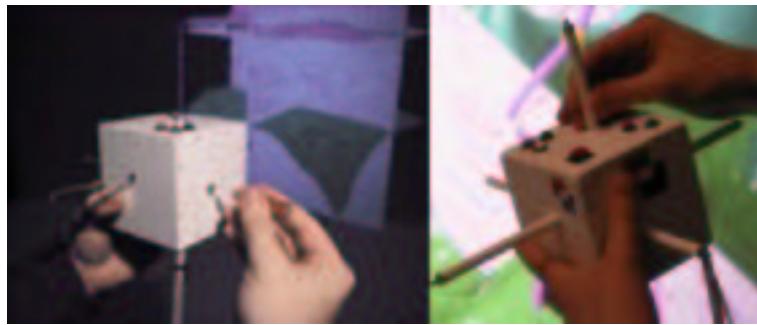


FIG. 2.5 – La *Cubic Mouse* de Frohlich et al. [46]

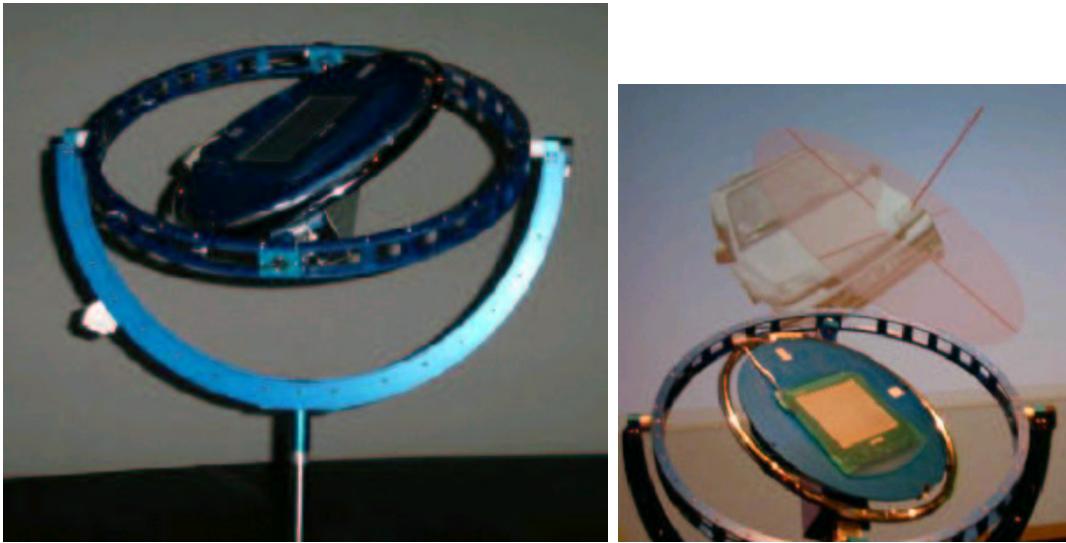


FIG. 2.6 – La table d'interaction de Hachet et Guitton [58]

lations. La table d'interaction favorise d'une part, par sa forme utile plane, des techniques d'interaction communes à celles de la palette, et d'autre part, par sa prise sous forme de disque ou volant, des techniques de rotation.

### 2.2.2.2 Equipements portés

Les mains sont dans le monde réel notre premier moyen d'interaction pour manipuler notre environnement. Aussi est-il naturel de retrouver en réalité virtuelle des dispositifs cherchant à exporter ce savoir-faire dans les environnements virtuels. Des gants peuvent être portés pour enregistrer les mouvements de la main. Un capteur repère la position du poignet. Deux grands types de gants existent : les gants de données [130] qui mesurent la position du poignet et la pliure des doigts par l'ajout de tiges flexibles sensibles à la

courbure, et les *pinch gloves* [41], littéralement gants de pincement, qui captent le contact entre l'extrémité du pouce et d'un autre doigt sous la forme d'un signal discret.

### 2.2.3 Conclusion

Dans un environnement immersif, l'interaction entre l'utilisateur et son environnement est directe. Pour agir, il doit pouvoir indiquer des positions et des orientations dans l'espace virtuel. Les équipements de l'interaction sont les intermédiaires physiques, portés ou tenus en mains, remplissant ce rôle. Nous avons vu qu'ils sont donc communément équipés d'un capteur de position et d'orientation, et peuvent prendre diverses formes. Celle-ci détermine très souvent la qualité de l'interaction future, du fait de sa forme utile et de sa prise pour les équipements tenus en main, ou de sa localisation sur le corps pour les équipements portés. Cette taxonomie des équipements physiques de l'interaction nous a permis de classer et de présenter quelques périphériques d'interaction courants de la réalité virtuelle.

## 2.3 Navigation

Ce paragraphe s'intéresse aux techniques de navigation c'est à dire aux actions réalisées par l'utilisateur pour se mouvoir à l'intérieur d'un espace virtuel. Le principal problème consiste à spécifier un déplacement virtuel quelconque sans déplacement physique ou avec un déplacement physique souvent restreint. Il faut aussi noter que le déplacement qui est réalisé dans le monde virtuel peut être plus complexe que son équivalent dans le monde réel du fait entre autres de l'absence de gravité qui autorise des déplacements 3D quelconques. Le matériel (interface matérielle, environnement virtuel, place disponible) peut contraindre les techniques utilisées. Une technique basée sur un mouvement des jambes ne peut pas être proposée à un utilisateur assis. De même qu'une technique utilisant une souris ne peut pas être proposée à un utilisateur debout sans plan de travail à sa disposition.

Plusieurs solutions ont été proposées pour répondre au problème du déplacement en environnement virtuel. On distingue les approches suivantes, réunies dans le tableau de

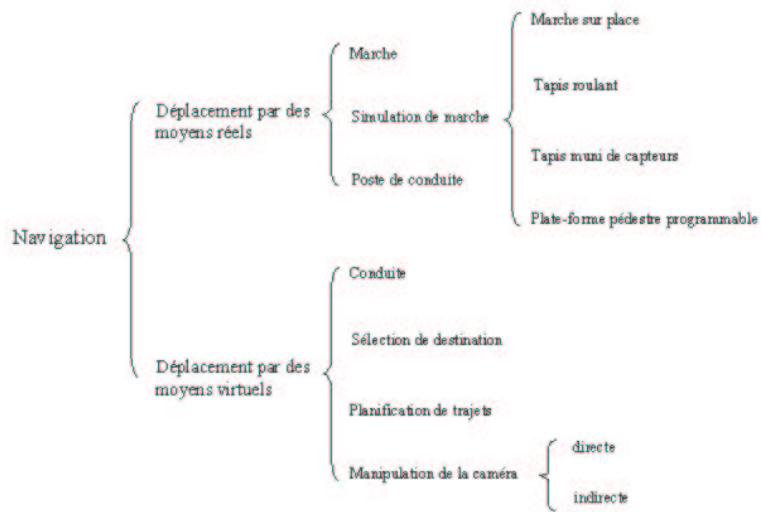


FIG. 2.7 – Taxonomie des méthodes de navigation

la figure 2.7.

### 2.3.1 Déplacement par des moyens réels

#### 2.3.1.1 Marche

La première approche pour permettre la navigation dans un environnement virtuel est de laisser l'utilisateur se déplacer dans l'espace de la manière la plus naturelle : en marchant. L'avantage est de permettre à l'utilisateur d'utiliser les sens kinesthésique (perception de la position, du mouvement et de la tension des muscles) et vestibulaire (perception de l'équilibre par l'oreille interne). Par contre, elle nécessite de disposer d'un espace de déplacement aussi large que l'espace virtuel. Cette technique requiert également le suivi des mouvements de la personne. Pour cela les solutions techniques sont diverses : suivi de mouvements magnétique, optique [37], ultrasons, etc.

#### 2.3.1.2 Simulation de marche

Pour contourner cette exigence d'espace, diverses solutions techniques ont été développées pour simuler la marche ou pour contraindre le déplacement réel de l'utilisateur pendant la marche. Hollerbach [64] présente en 2002 un état de l'art de ces interfaces

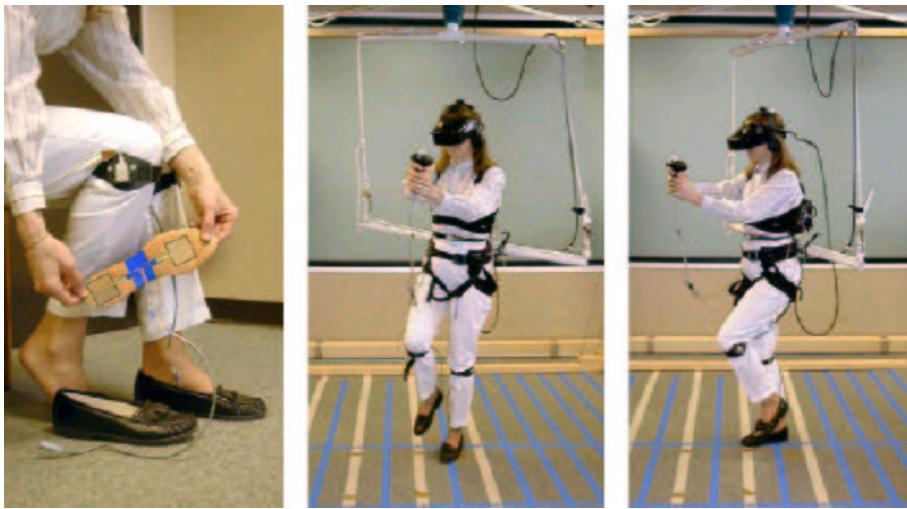


FIG. 2.8 – Le système GAITER (Templeman, Naval Research Labs)

contrôlant les déplacements. Les principales solutions abordées comprennent : la marche sur place, les tapis roulants, les tapis munis de capteurs et les plates-formes pédestres programmables.

**2.3.1.2.1 La marche sur place** Une première solution pour respecter la contrainte d'espace est de permettre à l'utilisateur de marcher sur place [116, 64]. Le projet GAITER (Templeman Naval Research Labs) est un système d'analyse de mouvements qui détermine la direction et la vitesse de déplacement dans l'univers virtuel à partir des mouvements de pieds enregistrés (cf. figure 2.8).

**2.3.1.2.2 Tapis roulants** L'utilisation de tapis roulants permet aussi de réduire l'espace nécessaire aux mouvements physiques de l'utilisateur (voir figure 2.9). Le sujet se tient debout et marche sur un tapis mobile. Plusieurs dispositifs de ce type ont été développés. Il existe des tapis monodirectionnels ou omnidirectionnels comme par exemple le tapis omnidirectionnel d'Iwata et al. (voir figure 2.10). Des tapis plus récents permettent de simuler des pentes, en s'inclinant [65].

**2.3.1.2.3 Tapis munis de capteurs** D'autres techniques utilisent un tapis non mobile muni de capteurs comme le VMC ou *Virtual Motion Controller* (Hit Lab Univ. Washington) (voir figure 2.11). Celui-ci se compose d'un tapis incurvé comprenant 4 capteurs



FIG. 2.9 – Tapis unidirectionnel, copyright Sarcos



FIG. 2.10 – Tapis omni-directionnel, copyright Iwata 1996



FIG. 2.11 – Le *Virtual Motion Controller*, un tapis muni de capteurs, copyright Human Interface Technology Lab, Washington

de poids grâce auxquels le système détermine la direction intentionnelle de l'utilisateur (voir figure 2.12). L'éloignement de la position centrale détermine la vitesse de déplacement. Le *Magic Carpet* (Georgia Tech) est semblable au VMC, il s'agit d'un tapis plan, recouvert de capteurs piézo-électriques permettant de mesurer la position des pieds (voir figure 2.13).

**2.3.1.2.4 Plate-forme pédestre programmable** Enfin, une dernière technique permet de marcher sur un terrain non plat et infini dans toutes les directions, à l'aide d'un dispositif complexe reliant les pieds de l'utilisateur à des plates-formes mécaniques, comme le système *Sarcos Bipart*, ou les *Gaitmaster 1* et *2* d'Iwata (cf. figure 2.14).

### 2.3.1.3 Poste de conduite

Une métaphore naturelle pour gérer la navigation consiste à utiliser un poste de conduite : poste de conduite automobile (avec un volant), poste de pilotage (avec un joystick) ou tout simplement une bicyclette [21]. La bicyclette est un moyen simple d'ac-

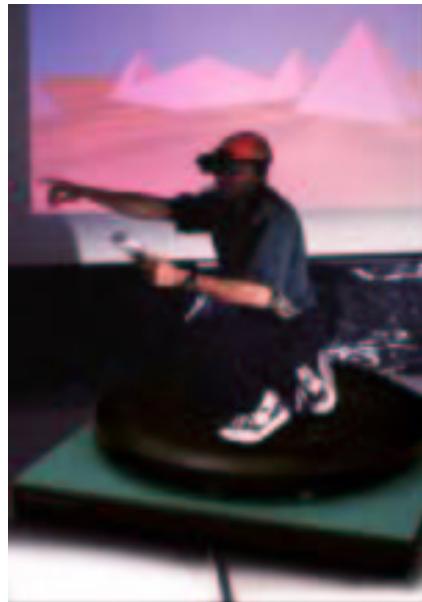


FIG. 2.12 – Tapis à capteurs de poids

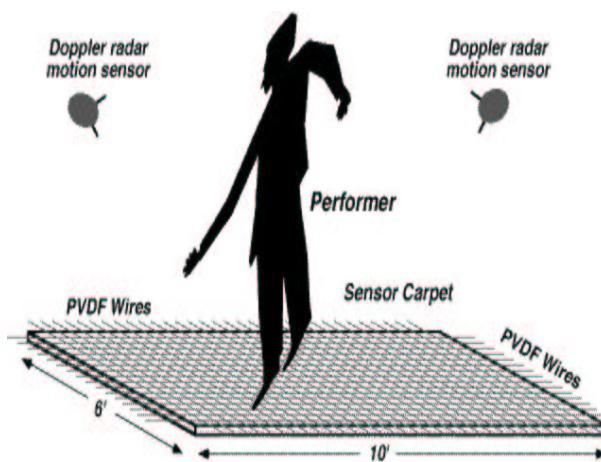


FIG. 2.13 – Magic Carpet, Tapis à capteurs piezo-électriques



FIG. 2.14 – Le système Gaitmaster 1 d'Iwata

quérir une intention de déplacement à 3 degrés de liberté (cf. figure 2.15). Celle-ci est utilisée pour simuler un déplacement par pédalage, et une commande de direction. Cette technique permet aussi un retour d'effort au travers des pédales et du guidon.



FIG. 2.15 – Déplacement utilisant une bicyclette fixe (Sarcos Uniport [103])

### 2.3.2 Déplacement par des moyens virtuels

#### 2.3.2.1 Conduite

Nous avons vu que la métaphore du poste de conduite pouvait servir à construire des équipements physiques pour se déplacer dans le monde virtuel. Elle est également un paradigme de déplacement populaire dans les environnements virtuels. La métaphore de la conduite suggère une gestion du déplacement par la spécification continue de la direction de mouvement. Les méthodes pour contrôler la direction de mouvement sont diverses :

- **par le regard** : Le système analyse la direction de regard de l'utilisateur ou par défaut les mouvements de la tête et leur associe un déplacement [92].
- **par pointage** : L'utilisateur désigne la direction de déplacement avec la main [91], le doigt, ou en orientant un outil dont la position est repérée par un capteur.

La vitesse de déplacement, les accélérations et décélérations doivent être contrôlées par des mécanismes connexes. Il est intéressant de noter que dans un environnement virtuel, l'utilisateur n'est pas contraint par des notions de gravité (sauf si elle est volontairement simulée) et peut donc suivre non seulement une métaphore de conduite automobile, c'est à dire de déplacement au sol, mais aussi une métaphore de conduite d'avion et voler dans toutes les directions de l'espace.

#### 2.3.2.2 Sélection de destination

Le but de la navigation est de permettre à un utilisateur de se déplacer d'un point à un autre, selon ses désirs. Plutôt que de contrôler à chaque étape le déplacement lui-même, cette technique propose de ne s'intéresser qu'à la destination de l'utilisateur. Celui-ci désigne une cible par une technique de sélection quelconque (voir ci-après la section sur la sélection/manipulation pour plus de détails) et le système déplace alors automatiquement le point de vue de l'utilisateur vers cet endroit. Deux autres méthodes courantes de sélection de destination consistent à choisir une position spatiale dans une liste ou à entrer des coordonnées.

Pour mieux planifier son déplacement par sélection de destination, il peut être sou-

haitable de visualiser et contrôler interactivement les étapes de son déplacement. Une méthode de planification de trajet consiste à disposer dans une vue globale et complète du monde virtuel des *waypoints* ou points de passage.

### 2.3.2.3 Manipulation de la caméra

**2.3.2.3.1 Manipulation directe** Une autre méthode offrant également une vue globale de l'univers et du déplacement que l'utilisateur souhaite effectuer, propose d'ajouter à la vue, devant l'utilisateur, une maquette miniature de tout l'univers. Dans ce "monde en miniature" [113] (*World In Miniature* ou WIM en anglais), une représentation de l'utilisateur est également insérée à la position exacte de celui-ci dans le monde virtuel macroscopique. Il peut déplacer son avatar dans le monde en miniature pour changer simultanément le point de vue devant ses yeux.

**2.3.2.3.2 Manipulation indirecte** L'utilisateur est immergé dans l'univers virtuel. Agir par le biais d'une vue extérieure au monde virtuel (une vue sur une maquette) pour contrôler la caméra, peut être moins naturel qu'une technique conservant l'attention de l'utilisateur sur la vue subjective du monde. La manipulation indirecte de la caméra conserve cette attention, en proposant d'agir sur des objets visibles, pour déplacer indirectement par répercussion le point de vue.

Dans la technique de manipulation d'objets fixes, l'utilisateur sélectionne et saisit un objet, qui reste fixe dans le monde. Lorsqu'il cherche à le déplacer, c'est lui même qui se déplace en contrepartie. Cette méthode peut rappeler des actions du monde réel : pousser sur les murs ou tirer sur un objet fixe pour s'en approcher.

Une méthode similaire fonctionne en agrippant l'air [8, 66, 85, 88]. L'espace entier est tiré vers l'utilisateur lorsqu'il ramène sa main vers lui. Il peut progresser ainsi par tractions successives avec les deux mains.

### 2.3.3 Conclusion

Un des buts de la réalité virtuelle est de produire des environnements virtuels les plus immersifs possibles. Il est par conséquent tentant de vouloir se déplacer dans un environnement virtuel par la méthode la plus naturelle : la marche. Les contraintes d'espace et les équipements portés, "attachant" l'utilisateur à des appareils fixes, interdisent souvent cette solution immédiate. Des installations complexes permettent d'autoriser la marche ou toute autre technique de déplacement courante et d'avancer ainsi dans l'univers virtuel, sans qu'il y ait pour autant de déplacement effectif de l'utilisateur dans le monde réel. A l'inverse, une approche virtuelle du problème permet de simuler la navigation dans un univers virtuel de manière peut-être moins naturelle, mais aussi moins coûteuse et moins complexe. Les paradigmes sont variés, favorisant un contrôle naturel, proche de situations réelles, favorisant la rapidité de déplacement, ou encore une vision globale du trajet pour planifier le déplacement.

## 2.4 Sélection et Manipulation

Dans les univers virtuels comme dans le monde réel, interagir avec son environnement passe par la faculté d'un individu à saisir, tourner, déplacer ou altérer des objets. Bien que ces tâches ressemblent à celles que nous pratiquons tous les jours sur notre environnement, elles se distinguent par la relation nouvelle que l'utilisateur doit développer avec des objets virtuels. En effet ces objets n'ont d'existence qu'à travers l'illusion générée par la stimulation artificielle des sens humains. Les sensations simulées bien que reconnaissables, restent fictives. Il est encore difficile d'agripper et de manipuler dans ses mains un objet virtuel. De plus il est possible dans un environnement virtuel de créer des objets ou des comportements qui n'ont aucun équivalent réel.

Cette forme d'interaction nouvelle, entre un être humain et un univers fictif s'intéresse à l'ensemble des tâches de manipulation possibles entre l'utilisateur et les objets de la scène virtuelle. Elle peut prendre de multiples apparences, selon le type d'objets à manipuler, sa disposition dans l'espace et l'intention de l'utilisateur. Afin de simplifier le problème, les concepteurs, qui doivent fournir des méthodes pour chacune de ces interactions souhaitées, se sont penchés en premier lieu sur la recherche de tâches canoniques pour la manipulation

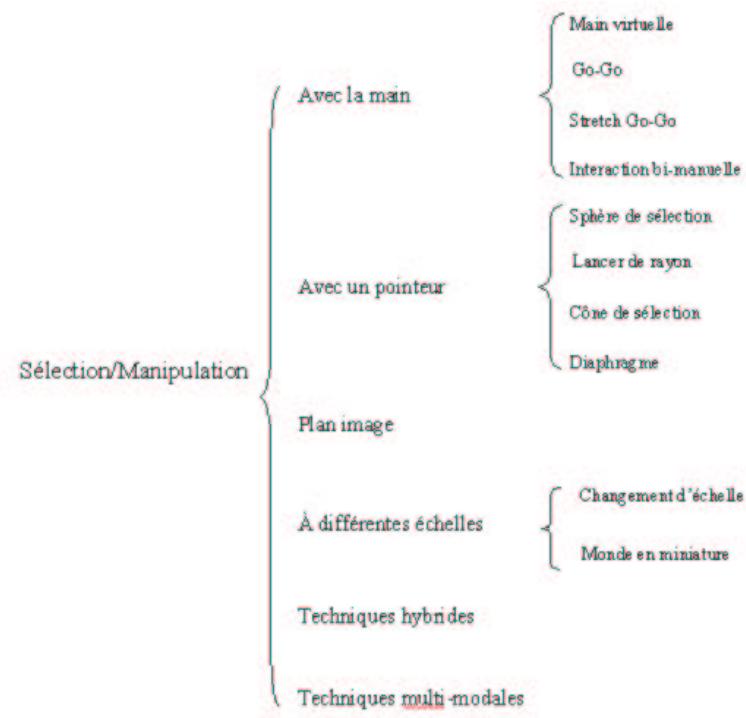


FIG. 2.16 – Taxonomie des méthodes de sélection/manipulation

d'objets rigides.

Si l'on se réfère aux travaux précédents sur les interfaces 2D [43, 51] et aux classifications d'interfaces 3D [87, 98] nous pouvons distinguer trois tâches élémentaires de manipulation :

- Sélection : la tâche d'identification manuelle d'un objet.
- Positionnement : la tâche de déplacement d'un objet depuis une position de départ vers une position finale.
- Rotation : La tâche de rotation d'un objet depuis une orientation de départ vers une orientation finale.

L'objectif des techniques de sélection/manipulation en environnement virtuel consiste à répondre à ces trois exigences pour l'interaction proche et distante, et pour des objets virtuels de nature et de taille quelconque. Des approches très diverses ont été suivies, se concentrant sur la forme utile du périphérique d'interaction (main, stylus), travaillant sur les dimensions mêmes de l'espace, combinant ces techniques entre elles ou encore explorant des modalités additionnelles, comme la voix. Le tableau de la figure 2.16 résume ces différentes approches.

## 2.4.1 Sélection-manipulation avec la main

### 2.4.1.1 Main virtuelle

Dans un environnement virtuel immersif l'utilisateur est plongé dans un univers 3D fictif. Ses sens sont stimulés pour lui donner l'illusion qu'il évolue et peut interagir avec les éléments de cet univers virtuel, comme il le fait dans le monde réel. Une méthode spontanée pour manipuler des objets 3D consiste à tendre le bras pour les attraper avec la main [115]. Selon la configuration de réalité virtuelle et les équipements disponibles, ce sera directement la main de l'utilisateur qui attrapera les objets ou un avatar la représentant dans l'univers virtuel (notamment pour les visio-casques avec la *virtual hand* de Poupyrev [99]).

Zhai et al. [128] soulignent l'intérêt d'employer l'ensemble de la main pour les rotations, c'est à dire obtenir les informations sur la position des doigts en plus du capteur sur la paume. Ken Hinckley [63] étudie différents périphériques pour la rotation d'objets 3D, dont un capteur industriel simple, et un capteur revêtu d'une coque sphérique pour faciliter la prise en main.

S'intéressant à l'amélioration des performances des techniques de rotation, Poupyrev [100] propose d'amplifier virtuellement les rotations d'un objet physique tenu en main pour produire des rotations non isomorphiques, avec les avantages suivants :

- Réduire le besoin de relâcher/reprendre l'objet que l'on tourne pour les rotations à large amplitude
- Etendre la gamme de valeurs couvertes par les dispositifs d'entrée pour la rotation
- Contrôler la sensibilité en rotation, de la plus rapide à la plus lente

Cette approche de la main virtuelle est assez intuitive et peut être satisfaisante pour les objets positionnés à portée de main. Cependant l'utilisateur est souvent confronté à un univers beaucoup plus grand, où il peut avoir besoin d'atteindre des objets plus distants. Il n'est pas toujours possible, simple, voire même préférable de se déplacer chaque fois que l'on veut atteindre un nouvel objet. Par conséquent il faut développer de nouvelles méthodes, inexistantes dans le monde réel mais possibles dans un univers virtuel, pour remplir ce double objectif de sélection et de manipulation d'objets à distance variable.

### 2.4.1.2 Go-Go

La technique dite du Go-Go, inventé par Poupyrev et al. [99] propose tout simplement d'étendre la portée de la main en utilisant un avatar pour la représenter et une fonction non linéaire de correspondance entre l'extension de la main réelle et celle virtuelle. L'espace autour de l'utilisateur se scinde alors en deux régions concentriques. Lorsque les mains de l'utilisateur évoluent dans la région la plus proche, c'est à dire lorsque la main reste à une distance de l'utilisateur inférieure à une certaine valeur seuil  $D$ , la main virtuelle suit les mouvements de la main réelle avec une correspondance un pour un. D'autre part, lorsque l'utilisateur étend ses mains au delà de cette distance  $D$ , la correspondance devient non-linéaire et le bras virtuel de l'utilisateur "croît", permettant ainsi à celui-ci d'atteindre et de manipuler des objets auparavant trop éloignés. Des évaluations ont montré que la technique reste assez naturelle. Malgré tout, l'utilisateur perd en précision au fur et à mesure que ses mains s'éloignent et que de petits mouvements de ses mains sont appliqués comme de grands déplacements dans l'espace virtuel. La distance maximale des objets atteignables et confortablement manipulables par cette technique n'est donc pas infinie.

### 2.4.1.3 Stretch Go-Go

Une variante de cette technique est proposée par Bowman [20] sous le nom de *stretch go-go*. Elle permet d'atteindre des objets à une distance infinie et de les manipuler comme s'ils étaient à portée de mains. L'espace autour de l'utilisateur est divisé en trois zones concentriques. Lorsque l'utilisateur étend son bras dans la zone la plus lointaine, l'avatar de son bras s'étend à une vitesse constante. Lorsqu'il ramène son bras dans la zone la plus proche de lui, son avatar se réduit à la même vitesse. Dans la zone médiane, la longueur de son bras virtuel reste constante et il peut manipuler des objets. Ce paradigme étend ainsi les possibilités du go-go tout en étant malgré tout moins naturel et moins direct. Un inconvénient de cette méthode est qu'elle induit une fatigue des bras, qui doivent maintenir des extensions partielles le temps des allongements ou des réductions de longueur des bras. Bowman cite également une technique alternative, consistant à contrôler l'allongement des bras par les boutons d'une souris 3D. Cette technique brise la cohésion de la méthode du go-go, mais pourrait être plus précise et physiquement moins fatigante.

#### 2.4.1.4 Travail bi-manuel

Le travail collaboratif des deux mains peut grandement accroître les performances en manipulation. La main non dominante n'est pas une simple version inférieure de la main dominante dans les tâches de manipulation [67]. Les travaux de Guiard [57] montrent notamment le rôle que jouent chacune des mains dans des tâches asymétriques et la hiérarchie naturelle qui s'instaure entre elles. La main non dominante est employée pour les tâches de positionnement global, ne requérant pas une précision particulière. Son rôle primordial est de servir de cadre de référence à la main dominante, qui peut alors effectuer des tâches de positionnement relatif, avec aisance et une grande précision. De nombreux auteurs se sont penchés depuis sur le travail bi-manuel et ses performances [2, 3].

Cutler et al. [33] explorent de nombreuses techniques de manipulation s'appuyant sur le travail conjoint des deux mains [125] (voir figure 2.17). Le premier paradigme, baptisé "table tournante" s'inspire de la forme de table du Plan de Travail Virtuel. L'utilisateur saisit deux points de l'espace en pincant deux doigts de chaque main. Le milieu du segment dessiné par ces deux points définit le point d'intersection de la table et de son axe de rotation. Il peut alors tourner le modèle virtuel reposant sur la table en déplaçant ses mains autour de l'axe. *Slide and Turn* (littéralement glisse et tourne) est une variante de la technique précédente, pour laquelle l'axe de rotation n'est pas fixe, mais reste contraint perpendiculairement à la table. Il permet de déplacer et de tourner le modèle sur la table. En retirant des contraintes supplémentaires Cutler et al. développent les techniques *grab and carry* (saisit et soulève) et *grab and twirl* (saisit et tord), permettant respectivement des mouvements à 5 et 6 degrés de liberté. Dans la première technique, l'axe de rotation n'est plus contraint et peut prendre n'importe quelle orientation. Les rotations le long de l'axe des deux mains, restent le seul degré de liberté non permis. Dans la dernière technique cette possibilité est octroyée en associant la rotation axiale à la torsion du poignet gauche ou droit de l'utilisateur.

Ils développent également une technique de grossissement fonctionnant sur le travail combiné des deux mains. La main non dominante détermine le point focal du zoom, et l'éloignement relatif de la main dominante règle le facteur de zoom (voir figure 2.18).



FIG. 2.17 – Translation et rotation à deux mains, copyright Cutler et al. 1997



FIG. 2.18 – Opération de grossissement à deux mains, copyright Cutler et al. 1997

## 2.4.2 Sélection-manipulation avec un pointeur

L'utilisation d'un outil tenu en main est une seconde approche au problème de la sélection-manipulation. Les équipements présentant une forme utile allongée, comme le *wand* ou le *stylus*, sont spécialement conçus pour pointer des éléments de la scène et donc inspirer des techniques de sélection. Pour ces tâches précisément, une étude suggère que l'utilisation du *stylus* permet une meilleure précision que l'utilisation de gants de données [106].

### 2.4.2.1 Sphère de sélection

Pour des objets à portée de bras et de petite taille, une sélection directe par le biais d'un objet physique, comme la pointe d'un stylo (équipé d'un capteur), n'est pas toujours évidente lors de tâches de précision, à cause des problèmes de correspondance entre les mondes réel et virtuel et la latence des systèmes. Raffaele de Amicis et al. [34] proposent au sein d'une application de CAD une technique de sélection fondée sur une sphère transparente apparaissant à l'extrémité du stylo physique. Zhai et al. [129] ont en effet montré les bénéfices de la transparence pour l'acquisition de cibles. La sphère transforme l'extrémité du stylo, point unique de l'espace, en petit volume de sélection. Elle permet notamment à l'application de déterminer si l'utilisateur souhaite désigner un sommet, une arête ou une face de son modèle. Lorsque plusieurs objets sont reconnus dans le volume de la sphère, la priorité de sélection est donnée par l'ordre topologique (sommet, arête et enfin face), puis par la proximité géométrique.

### 2.4.2.2 Lancer de rayon

Suivant une approche différente, la technique *Ray-casting* ou lancer de rayon [13] consiste à utiliser un rayon virtuel prolongeant la main ou un objet tenu en main, pour pointer directement sur les objets de la scène (à la manière d'un rayon laser). Cet outil pour pointer est naturel et s'utilise très simplement pour des tâches de sélection. La manipulation d'objets n'est cependant pas parfaite, le rayon ne permettant pas de réaliser simplement les rotations et certains déplacements. Il peut être également difficile de saisir les objets petits ou distants, du fait du mode d'interaction, indirect. Depuis la

première implémentation de cette technique par Bolt dans le système *Put-That-There* de très nombreuses variantes de cette méthode de lancer de rayons ont vu le jour [84].

Une des difficultés inhérentes à la plupart des techniques de manipulation d'objets lointains concerne le contrôle de la distance de l'objet à l'utilisateur. La sélection par rayon présente cette même faiblesse : il est très malaisé de déplacer un objet selon l'axe de visée. Un méthode pour adoucir cette difficulté consiste à s'inspirer du moulinet d'une canne à pêche (*fishing reel* en anglais) [20]. Après avoir sélectionné un objet l'utilisateur peut l'enrouler ou le dérouler le long de la ligne en utilisant la souris ou tout autre périphérique dédié. Ce moulinet de pêche permet à l'utilisateur de régler la distance de l'objet mais introduit une séparation des différents degrés de liberté en translation.

#### 2.4.2.3 Cône de sélection

JDCAD est un système de modélisation développé par Liang en 1993 [78] pour lequel il propose une nouvelle technique de sélection d'objets améliorant le système de projection de rayon. Le système fonctionne avec un moniteur standard et un simple capteur magnétique similaire à la *bat* de Ware [24], tenu en main. La tête est repérée dans l'espace à l'aide d'un second capteur.

La *bat* projette un cône virtuel, à la manière d'un faisceau de lampe torche. En orientant le capteur, l'utilisateur peut "éclairer" différents objets. Tous les objets pris dans le cône sont des candidats à la sélection (voir figure 2.19). Cette méthode permet de sélectionner facilement les objets, qu'ils soient de petites tailles ou éloignés de l'utilisateur. En contrepartie, lorsque plusieurs objets sont pris dans le cône de sélection, il n'est pas possible de choisir l'objet sélectionné, à moins de lui adjoindre des techniques supplémentaires pour lever l'ambiguité.

#### 2.4.2.4 Diaphragme

La technique du diaphragme (*aperture* en anglais) de Forsberg et al. [44] est une modification de la technique précédente, qui permet à l'utilisateur de contrôler interactivement le volume de sélection. La direction du cône est calculée à partir de la position de la tête, estimée à partir des valeurs d'un capteur porté sur la tête, et la position d'un capteur

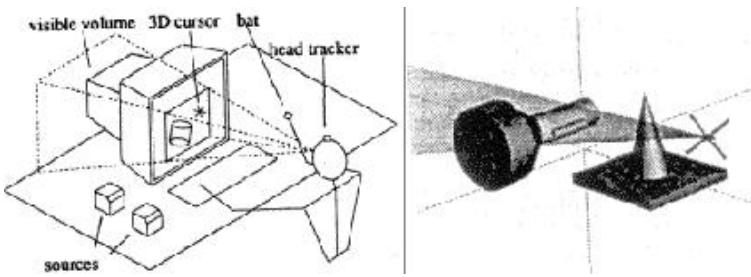


FIG. 2.19 – Le cône de sélection dans le système de modélisation JDCAD

tenu en main. La distance entre la main et la tête de l'utilisateur fait varier l'ouverture du cône de sélection à la manière d'un diaphragme d'appareil photographique. La technique du diaphragme améliore la technique de la lampe torche, en fournissant un mécanisme interactif pour lever l'ambiguité en affinant le cône de sélection.

### 2.4.3 Plan image

Une troisième famille de méthode de sélection, appelée *Image plane* [95], s'appuie sur l'idée de manipuler une projection 2D de l'espace 3D plutôt que l'espace 3D lui-même. Cette simplification réduit le nombre de degrés de liberté et ramène le problème à une sélection en deux dimensions. L'utilisateur désigne les objets en pointant du doigt sur leur projection sur un plan image 2D (voir figure 2.20). L'objet sous le doigt est sélectionné en considérant le vecteur partant de l'œil vers ce point, et son intersection avec les éléments de la scène.

Cette sélection peut également être étendue à des objets proches en utilisant la deuxième main pour définir le rayon d'un cercle englobant [96] (figure 2.21).

Alternativement l'utilisateur peut sélectionner des objets avec deux doigts, en positionnant son pouce et son index en dessous et au dessus de l'objet désiré (technique baptisée *Headcrusher*, cf. figures 2.22 et 2.23). Après sélection, la "projection" de l'objet peut être manipulée : l'objet est réduit et ramené à portée de l'utilisateur, de telle manière qu'il puisse en quelque sorte manipuler la projection. Quoi qu'il en soit, la distance de l'objet ne peut être contrôlée directement par cette méthode.

Une approche semblable de Schmalstieg et al. [104] propose de matérialiser le plan

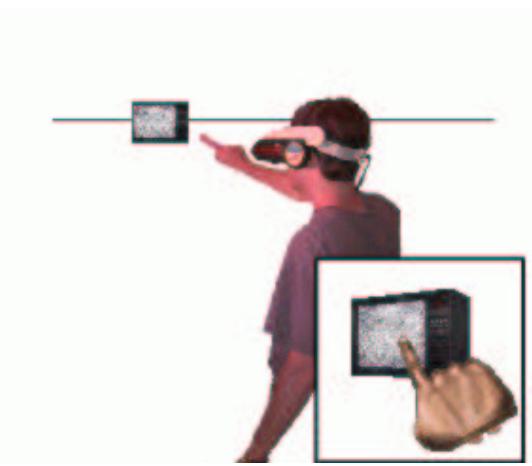


FIG. 2.20 – La technique *Sticky Finger* pour attraper un objet virtuel distant



FIG. 2.21 – La technique "Sticky Finger" élargie par un cercle de sélection de rayon variable



FIG. 2.22 – Principe du *Headcrusher*



FIG. 2.23 – La technique *Headcrusher* pour attraper un objet virtuel distant

en tenant en main une plaque transparente. La sélection est réalisée en centrant l'objet désiré au milieu de la plaque ou à l'aide d'un stylo en entourant d'un cercle approximatif les éléments qui apparaissent par transparence sur la plaque.

#### 2.4.4 Sélection-manipulation à différentes échelles

##### 2.4.4.1 Changement d'échelle

Mine propose de contourner les difficultés de la manipulation d'objets distants, en rapprochant les objets de l'utilisateur par un changement de l'échelle du monde [88]. Une position médiane entre les deux yeux de l'utilisateur est choisie comme centre de cette homothétie sur les objets du monde. Les objets et distances du monde diminuent (où l'utilisateur grandit, selon le point de vue adopté) et rapprochent l'objet pointé à portée de main de façon plus ou moins transparente, en conservant l'effet de perspective. Contrairement à la technique précédente du plan image, l'utilisateur peut cette fois contrôler la profondeur des objets et ses rotations, puisqu'il est libre de les manipuler dans toutes les directions de l'espace. Ses mouvements sont cependant amplifiés par le facteur d'échelle et réaliser de petits déplacements est plus ou moins aisé, selon la distance initiale de l'objet attrapé.

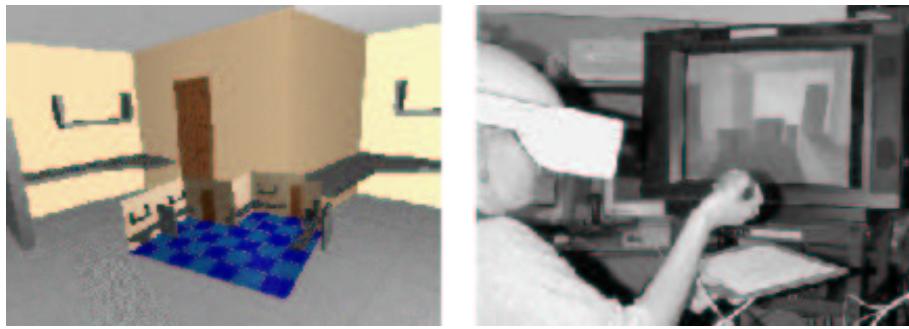


FIG. 2.24 – Monde en miniature

#### 2.4.4.2 Monde en miniature

Une autre alternative pour étendre la portée de l'utilisateur est de réduire le monde entier et l'amener à distance de bras. La technique du monde en miniature, *World-In-Miniature* ou WIM en anglais [113], présente à l'utilisateur un modèle miniature tenu en main de l'environnement virtuel, réduit selon un coefficient constant (voir figure 2.24). L'utilisateur peut ainsi manipuler indirectement les objets virtuels en agissant sur leur représentation miniature dans le WIM. La technique WIM est une technique puissante qui combine une manipulation d'objets distants et proches avec une technique de navigation, puisque l'utilisateur peut être représenté lui-même dans le modèle miniature et déplacé selon ses souhaits. Cependant, la technique se prête mal au contrôle de petits objets, qui deviennent très difficiles à manipuler une fois réduits dans le WIM.

#### 2.4.5 Techniques de manipulation hybrides

Les techniques précédentes peuvent également être combinées pour donner naissance à des techniques hybrides plus performantes.

Bowman, après avoir étudié certaines de ces méthodes de sélection et de manipulation, développe une technique baptisée HOMER [20] (pour *Hand-centered Object Manipulation Extending Ray-casting* en anglais). La sélection des objets est obtenue par l'utilisation simple et intuitive d'un rayon virtuel. Dès qu'un objet est sélectionné, la main virtuelle apparaît auprès de lui pour pouvoir le manipuler. La manipulation s'appuie sur la distance de l'objet à l'utilisateur et la distance de sa main à son corps. Il peut ainsi tout à la fois déplacer l'objet sur une sphère centrée sur lui, et contrôler indépendamment les rotations

de l'objet lui-même. Le contrôle de la profondeur est associée à deux versions différentes de la technique HOMER : directe ou indirecte. Dans la méthode directe, la position de l'objet est couplée à l'extension courante de la main au moment où il est saisi. Si l'utilisateur étend son bras, l'objet s'éloigne, s'il le rapproche, l'objet fait de même. Dans la version indirecte d'HOMER, la distance est contrôlée avec plus de finesse et sans limite par les boutons d'une souris 3D.

La technique *Voodoo Dolls* [96] de Pierce et al. propose une interaction naturelle à deux mains pour manipuler et positionner des objets relativement les uns aux autres. Chaque main saisit les objets à manipuler par la méthode du plan image. La main non dominante fixe le référentiel en tenant le premier objet et la main dominante travaille dans ce référentiel pour placer le second objet relativement au premier, selon les recommandations de Guiard [57]. L'objet de référence tenu en main non dominante est remis à l'échelle pour mesurer un demi mètre dans sa plus grande dimension, tandis que l'objet en main dominante est transformé en conséquence pour respecter les proportions.

#### 2.4.6 Techniques de manipulation multi-modales

Le système *Put-That-There* de Bolt [13], littéralement : Pose-Ceci-Ici, combine la voix et le geste pour manipuler des objets sur une carte. Chacune de ces deux modalités, prise isolément, est limitée. La reconnaissance vocale n'est pas un outil encore fiable à 100%. Il peut être à la fois fatigant et aléatoire de devoir décrire chacun des objets que l'on désire manipuler. Le geste (désigner du doigt) est efficace pour sélectionner et déplacer des objets, mais guère plus. L'idée principale consiste à combiner ces deux média pour combler réciproquement leurs faiblesses et construire un nouveau mode de communication plus naturel et direct. A la place de dire "pose le cercle rouge en bas à droite du petit carré bleu", l'utilisateur dira "pose ceci, ici" en pointant les éléments concernés. Aucun objet n'a besoin d'être décrit, aucun nom d'objet n'a plus besoin d'être mémorisé. Chacun d'eux est décrit par "ceci". Cela simplifie également la syntaxe des commandes vocales.

Hauptmann étudie en 1989 [60] le comportement d'utilisateurs confrontés à des tâches de manipulation 3D : translation, rotation et changement d'échelle d'un cube, par le biais de trois modalités : geste, parole, et une combinaison des deux. Il en tire des indices sur l'efficacité d'une communication multi-modale. Les sujets de son expérience préfèrent

fortement utiliser geste et parole pour la réalisation des tâches demandées. Plus spécifiquement, la parole seule est jugée trop limitative pour fixer l'ensemble des paramètres des opérations. Le vocabulaire de ces commandes gestuelles et orales se révèle également étonnamment compact.

Cohen [29] expose les avantages de la parole pour la description et ceux du geste pour la manipulation directe. Les deux techniques peuvent ainsi se compléter harmonieusement pour compenser les faiblesses de l'autre. Par exemple, la voix n'est pas reliée à une métaphore spatiale et permet l'interaction avec des objets hors de vue, contrairement à l'interaction gestuelle.

Bolt et Herranz [12] explorent en 1992 une interaction basée sur la parole, les deux mains, et le regard. L'analyse de la direction de regard sert à déterminer quels objets doivent être sélectionnés. Les commandes vocales aident à lever les ambiguïtés sur les informations de regard et de désignation par geste. L'interaction à deux mains comprend les rotations et le placement relatif d'un objet par rapport à un autre, une main servant de référence à l'autre.

Sparrell et Koons [68, 69] développent un système utilisant des gestes iconiques et la voix. Ces gestes accompagnent et se réfèrent au contexte verbal, comme par exemple dans la phrase : "l'avion plane comme ceci".

D'autres études ont depuis confirmé que la voix et les gestes peuvent se compléter harmonieusement et efficacement pour augmenter la qualité de la communication homme-application [45, 111].

#### 2.4.7 Conclusion

Nous avons vu dans cette partie sur les techniques de sélection-manipulation plusieurs grandes familles d'approches, employant soit la main soit un outil pour pointer, et présentant chacune des avantages et des défauts selon la taille et la distance des objets, ou encore selon que l'accent soit placé sur la sélection ou sur la manipulation, les translations ou les rotations. Nous résumons ici les grandes classes de difficultés rencontrées par ces techniques :

- Sélection :
  - Petits objets : difficulté à sélectionner les objets de petite taille dans la scène.
  - Objets distants : difficulté à sélectionner les objets lointains, c'est à dire en général hors de portée de bras et pouvant être théoriquement à des distances infinies.
  - Ambiguité : la méthode de sélection ne doit pas fournir plusieurs candidats, sans procurer en même temps des moyens de lever l'ambiguité de la sélection.
- Manipulation
  - Aucune/2D/3D : la technique de sélection permet-elle également de manipuler les objets ? Si c'est le cas, est-elle restreinte en nombre de degrés de liberté pour la manipulation (notamment la gestion de la profondeur) ?
  - Objets distants : difficulté à manipuler les objets lointains, c'est à dire en général hors de portée de bras et pouvant être théoriquement à des distances infinies. Ces faiblesses peuvent notamment concerter la sensibilité trop aiguë des mouvements, ou à l'opposé la lenteur et la fatigue induite par des méthodes ne permettant pas des déplacements rapides sur de grandes distances.
  - Rotations : difficultés générales à réaliser des mouvements de rotation. Les points notables portent sur l'impossibilité de choisir le centre des rotations une fois l'objet sélectionné et sur l'amplitude limitée des rotations permises.

Il n'existe pas à ce jour de technique parfaite, idéale, et satisfaisant à tous les critères. Elles doivent être choisies judicieusement, en considérant leurs avantages et défauts respectifs et selon les objectifs des applications.

## 2.5 Contrôle d'application

Les premières applications pour environnements virtuels ont été confrontées au besoin immédiat de fournir une interface entre l'utilisateur et l'application. En plus de manipuler des données et des objets 3D, l'utilisateur doit pouvoir communiquer avec l'application elle-même pour appeler des commandes, changer de modes, modifier des paramètres ou déclencher des fonctions. Le contrôle d'application dans les environnements virtuels est un domaine de recherche jeune. Les premières applications pour ces environnements ont suivi une approche pragmatique, essayant de répondre rapidement mais souvent sommairement à leurs besoins particuliers. Différentes techniques d'interaction ont été ainsi développées

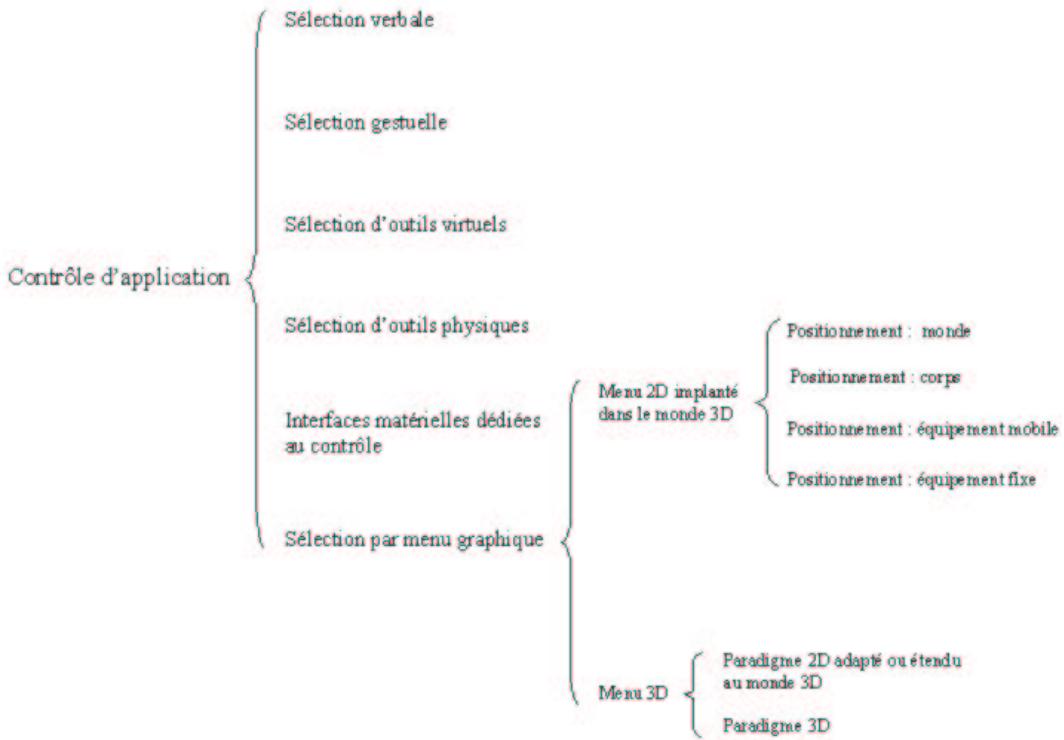


FIG. 2.25 – Taxonomie des méthodes de contrôle d'application

indépendamment les unes des autres.

Au cours des dernières années, plusieurs taxonomies [98, 19] ont été présentées, cherchant à organiser les différentes pistes explorées par ces techniques de contrôle. Beaudouin [5] analyse en 2000 l'évolution des interfaces du monde 2D vers les univers immersifs 3D. Il décrit un nouveau modèle d'interaction qu'il nomme interaction instrumentale, qui étend et généralise les principes de la manipulation directe. Ce modèle lui permet de décrire aussi bien les anciennes interfaces 2D que certaines nouvelles interfaces pour environnement virtuel. Lors d'un cours sur la conception des interfaces 3D [17], Ernst Kruijff, Joseph Laviola, Yvan Poupyrev, Doug Bowman et Mark Mine proposent une classification pour les techniques de contrôle d'application actuelles, influencée par la description de techniques non conventionnelles de MacMillan et al. [101]. Le premier critère de classification choisi est donné par le mode de sélection. On distingue principalement six modes de sélection : verbale, graphique, gestuelle, sélection d'outils virtuels et physiques, et utilisation d'interfaces matérielles dédiées au contrôle (voir figure 2.25).

Dans la suite nous décrivons chacune de ces six approches. Nous approfondirons plus

spécifiquement les menus à interface graphique, en adéquation avec les travaux de la thèse, pour lesquels nous proposons une classification, reprise dans le *Traité de la Réalité Virtuelle*, seconde édition [48].

Nous verrons dans une dernière partie, un domaine périphérique du contrôle d'application, concernant la saisie de chaînes de caractères en environnement virtuel. Cette tâche d'entrée de données, triviale sur station de travail, suscite encore de nombreuses difficultés dans les environnements virtuels.

### 2.5.1 Sélection verbale

Le mode verbal est attrayant à plusieurs égards : il est naturel, il laisse les mains libres, il ne requiert pas l'affichage de menus, ni l'utilisation d'interfaces matérielles qui peuvent être incompatibles avec la réalisation d'autres tâches. Il est cependant jusqu'à présent le moins répandu car, en contrepartie des avantages précédents, il est fatiguant pour l'utilisateur, il est difficilement compatible avec un travail de groupe où plusieurs personnes discutent ensemble, il nécessite un apprentissage qui peut devenir long si le nombre de commandes est important, et le vocabulaire reconnu par le système de reconnaissance vocale est généralement limité.

Weimer et al. [121] complètent l'interaction gestuelle d'un système CAD par des commandes vocales brèves ("ok", "fermer", "sortir") pour la confirmation ou le déclenchement de fonctions. Forsberg et al. [45] suivent la même approche pour leur système de modélisation sur Plan de Travail Virtuel. Ils notent l'intérêt particulier d'une seconde modalité, la voix, lorsque les deux mains sont occupées.

Parmi les applications de visualisation scientifique dans des configurations de réalité virtuelle le système Virtual Director [36] combiné à la librairie CAVE5D [26] permet l'utilisation complémentaire de la voix et du périphérique *Wand* tenu dans une main. Laviola [76] propose un dispositif similaire nommé MSVT (*Multimodal Scientific Visualization Tool*), remplaçant le *wand* par l'utilisation des deux mains équipées de gants, mais gardant la voix pour déclencher les commandes de l'application.

### 2.5.2 Sélection gestuelle

Dans le mode gestuel, une commande est associée à chaque geste ou posture [14]. Par abus de langage, on utilise le terme gestuel aussi bien pour la commande par postures (positions statiques) que par gestes (inclus le mouvement) comme par exemple le langage des signes. Il existe principalement deux technologies pour analyser les gestes ou les postures : la vision à l'aide de caméras, et l'utilisation de gants de données ou de *pinch gloves*. L'attrait de cette approche réside dans sa relative indépendance par rapport au système de visualisation (elle ne requiert pas l'affichage de menus ni l'utilisation d'interfaces matérielles de sélection encombrantes). Par contre, elle n'est pas très naturelle et l'utilisateur peut avoir des difficultés à mémoriser un grand nombre de commandes. Le port des gants peut aussi procurer une gêne pour les mouvements de la main et surtout "attache" l'utilisateur au système de suivi des mouvements à cause des câbles nécessaires à la connexion du système. L'acquisition des gestes par vision [90, 91] résout ces problèmes mais requiert des conditions d'éclairement qui ne sont pas toujours compatibles avec les systèmes de visualisation.

Il est intéressant de noter que si les modes vocal et gestuel ne requièrent pas d'affichage de menu, l'affichage d'un "pense-bête" rappelant les commandes et leur équivalent oral ou gestuel peut être très utile pour palier aux problèmes de mémorisation. Baudel propose un système de notation sous forme de schémas représentant les gestes de la main [4]. Mine [88] suggère de relier certains gestes véhiculant un contenu sémantique à des fonctions de l'application. Il propose ainsi de déclencher l'opération de destruction d'un objet lorsqu'un utilisateur déplace celui-ci au bout de sa main jusque dans son dos.

### 2.5.3 Sélection d'outils virtuels

L'outil virtuel est une métaphore qui rend le contrôle d'application très intuitive. L'outil est un concept présent dans de nombreux corps de métiers. A titre d'exemple, le menuisier dispose d'un certain nombre d'outils (ciseau à bois, rabot, scie, marteau,...) chacun étant réservé à un usage précis. La sélection de la tâche à accomplir (scier, enfoncer un clou, raboter,...) se fait par sélection de l'outil correspondant. De façon similaire, l'interface virtuelle propose un certain nombre d'outils (sous la forme d'outils virtuels) qui ont chacun une fonction précise. Ces outils peuvent être réunis sur un établi virtuel

comme le *toolbox* de Cutler [33]. La sélection de l'un ou l'autre de ces outils assure le choix de la commande. Notons ici que la frontière entre le menu graphique 3D et la sélection d'outils virtuels est un peu floue, les deux approches pouvant, dans certains cas, se rejoindre.

### 2.5.4 Sélection d'outils physiques

La sélection et l'emploi d'outils physiques est une approche conceptuellement similaire à l'utilisation d'outils virtuels mais, au lieu d'être proposés sous forme virtuelle, les outils sont proposés sous la forme d'accessoires, (ou *props* en anglais). Le fait de tenir un objet physique en main procure une sorte de "retour haptique passif" au niveau de la prise en main qui permet un usage plus intuitif suivant le rôle de l'outil. Un des exemples, parmi les plus significatifs, d'utilisation d'outils est l'application de neurochirurgie proposée par Hinckley et al. [62]. Dans cette application, le praticien dispose de plusieurs outils pour visualiser et manipuler son modèle (tête de poupée pour la manipulation du cerveau, plaquette de verre pour positionner le plan de coupe. Voir figure 2.4).

### 2.5.5 Interfaces matérielles dédiées au contrôle

L'avant dernière alternative consiste à employer une interface matérielle dédiée au contrôle d'application. On peut citer principalement deux approches :

- Le bouton est un mode de sélection très utilisé en 2D grâce aux touches du clavier ou aux boutons de la souris. Malheureusement, les contraintes de portabilité qui prédominent généralement dans un environnement virtuel ne permettent souvent pas de disposer d'interfaces matérielles avec de nombreuses touches. Comme nous l'avons vu dans la section sur les équipements physiques de l'interaction, les interfaces matérielles les plus courantes disposent généralement seulement d'un ou deux boutons. On peut cependant trouver quelques tentatives plus ou moins encombrantes et plus ou moins portables. Dans ce cas, la sélection d'une commande se ramène à une opération du type "raccourci clavier".

- Le PDA (*personal digital assistant*, un agenda électronique proche de l'ordinateur personnel tenu en main) peut être utilisé comme interface matérielle de contrôle. Il pro-

cure une puissance de calcul et une souplesse difficilement égalables et communique avec l'ordinateur hôte par infra-rouge. Watsen et al. utilisent par exemple un PalmPilot dans une CAVE [120].

### 2.5.6 Sélection graphique

Les menus graphiques forment la technique de contrôle traditionnelle des utilisateurs de station de travail. Dans les environnements virtuels, l'espace d'affichage est beaucoup plus vaste, puisque l'univers virtuel entoure l'utilisateur. Il est naturel, dès lors de profiter de ce large champ de vision pour disposer dans la scène des menus graphiques servant au contrôle d'application.

Le placement de ces menus graphiques à l'intérieur de l'univers virtuel est un sujet délicat. Il est important de trouver un compromis entre les considérations d'accessibilité, favorisant des menus volumineux et présents en permanence à la vue, et celles de discréption, favorisant des menus couvrant le moins possible et le moins longtemps la vue de la scène virtuelle. Différentes solutions existent pour disposer et/ou déplacer ces menus dans la scène [42]. Nous distinguons dans la suite quatre méthodes principales. Les menus peuvent être placés librement dans le monde virtuel (référence : monde), liés à une partie du corps de l'utilisateur comme la tête ou la main (référence : corps), connectés à un objet physique tenu en main (référence : équipement mobile), ou placés en référence à un objet physique fixe comme les bords du Plan de Travail Virtuel (référence : équipement fixe).

Les concepteurs d'interfaces immersives ont tout d'abord essayé d'importer les techniques de contrôle d'application connues et maîtrisées des stations de travail, dans l'univers virtuel. Dans cette approche, les paradigmes du monde 2D sont implantés directement dans l'univers 3D, tels quels. Elle permet de réutiliser les résultats des interfaces 2D et de proposer une interface familière. Cependant les périphériques d'interaction ne sont plus le clavier et la souris, l'interaction est directe et à trois dimensions. Des difficultés peuvent survenir du fait de ces nouvelles contraintes.

Une seconde famille de techniques s'intéresse plus spécifiquement aux possibilités et aux contraintes spécifiques des univers virtuels et cherche à proposer des paradigmes répondant aux défis d'une interaction directe et immersive. Ces nouveaux paradigmes

d'interaction peuvent être regroupés en deux catégories : les paradigmes inspirés du monde 2D, qui sont adaptés ou étendus à une troisième dimension, et les paradigmes purement 3D, qui n'ont pas d'équivalents dans le monde 2D.

#### 2.5.6.1 Menus 2D implantés dans le monde 3D

Une première approche naturelle pour les concepteurs d'interface graphique 3D a consisté à réexploiter les systèmes existants sur les stations de travail standards. L'univers 3D est projeté et affiché sur un écran à deux dimensions et l'interaction est indirecte et s'appuie sur deux périphériques : le clavier et la souris. Dans cet univers 2D une interface graphique s'est rapidement imposée : l'interface WIMP (*Windows, Icons, Menus and Pointing*). Elle propose une représentation de l'information et des techniques d'interaction construites sur la métaphore du bureau, où l'écran de travail est assimilé à un bureau virtuel sur lequel repose des documents. Des icônes et des fenêtres leur sont associés. Ces éléments sont déplaçables grâce à la souris, qui peut les saisir en cliquant, comme une main attrapant un dossier. Des menus linéaires à l'intérieur des fenêtres permettent de contrôler les applications. Cette approche, ancrée dans une réalité à deux dimensions, est devenue le standard de facto des interfaces graphiques pour stations de travail.

Puisque cette interface est populaire et assez efficace pour contrôler son univers 2D, une première approche consiste à porter directement l'interface WIMP dans le monde 3D [1]. Le principal avantage est de proposer aux utilisateurs un environnement de contrôle familier, donc intuitif, qui nécessitera un faible temps d'apprentissage. D'autre part, ce paradigme a été maintenant longuement étudié et dispose de nombreuses techniques de contrôle qui lui sont propres, comme les menus déroulants, les menus contextuels, les boîtes de dialogue, etc. qui peuvent être reprises pour enrichir l'interaction au sein d'un univers 3D.

Les menus linéaires à deux dimensions peuvent être intégrés de manière immédiate comme des éléments plats, ou avec un léger relief, flottants dans l'espace 3D. Cependant la métaphore du bureau, pensée pour un affichage sur un écran plat et des techniques d'interaction propres à un univers à deux dimensions, est moins adaptée à un environnement immersif 3D. La sélection d'options de menu dans un environnement 3D est notamment assez différente de la sélection en 2D. Sélectionner un objet parmi un ensemble de fonctions

est conceptuellement un choix à une seule dimension. L'ajout de dimensions supplémentaires est inutile et nuit à la simplicité de la tâche, ralentissant notablement le processus de sélection, comme le montre la loi de Fitt étendue à plusieurs dimensions [83]. Ainsi, pour des menus linéaires, plats, flottants dans l'espace 3D, de nouveaux problèmes de précision en profondeur peuvent apparaître.

**2.5.6.1.1 Positionnement : libre dans le monde** Les premiers menus 2D implantés dans le monde 3D se sont contentés de proposer des fenêtres flottant dans l'espace devant l'utilisateur. La sélection directe d'objets par un outil de pointage non contraint dans l'espace 3D n'est pas optimale [81]. Simuler des surfaces de contrainte [82] peut légèrement atténuer cette difficulté. L'utilisation d'un rayon virtuel [117] contrôlé par la main comme outil de pointage permet d'éliminer la contrainte de sélection en profondeur. Cependant la manipulation reste lente, car l'interaction n'est plus directe [93].

**2.5.6.1.2 Positionnement : centré sur un équipement mobile** Une autre approche consiste à attacher les fenêtres à un équipement physique. Wloka présente ainsi en 1995 un nouveau périphérique appelé *virtual tricorder* [124]. Cet équipement tenu en main se présente comme une souris à trois boutons, dont la position est repérée dans l'espace par un capteur. Wloka suggère d'ancrer des menus 2D sur le *virtual tricorder*. La fenêtre 2D est affichée comme un plan, à la position courante de l'équipement tenu en main (voir figure 2.26). Le déplacement et la sélection à l'intérieur du menu sont effectués en pressant les boutons de la souris.

Une technique de placement similaire propose de matérialiser le plan de sélection en tenant une palette [31, 81, 104] physique transparente et plate dans la main non dominante et de sélectionner les options avec la main dominante. Le menu graphique est affiché directement sur la vitre transparente (à l'aide d'un capteur sur la palette), tirant avantage d'une manipulation basée sur un outil physique offrant un retour d'effort passif et de l'affichage virtuel. En contrepartie, cette méthode constraint l'utilisateur à garder en permanence un équipement plus ou moins encombrant dans sa main, ou à portée de main, pour pouvoir accéder au menu. Pour des systèmes de réalité virtuelle à faible champ de vision, Blasko [9] développe un menu condensé sur un TouchPad<sup>TM</sup>, une palette sensible au toucher.

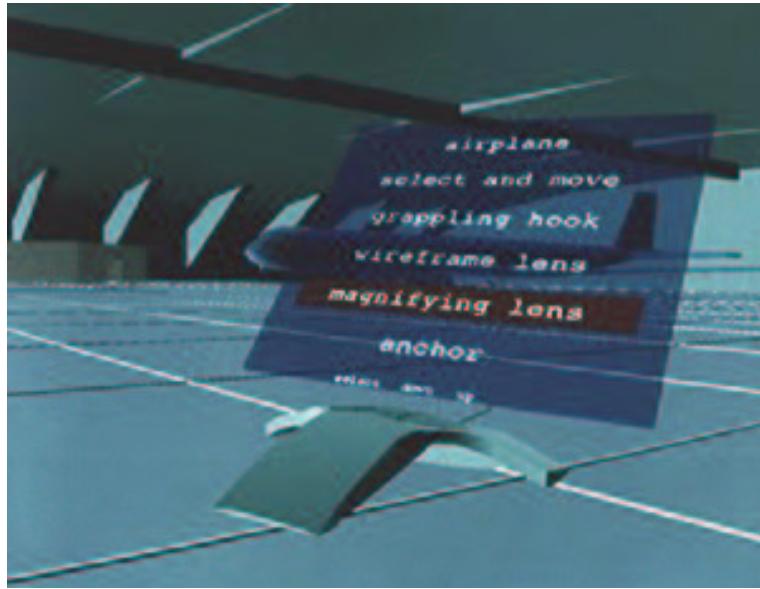


FIG. 2.26 – Le *virtual tricorder* contrôle la position des fenêtres et permet de naviguer parmi les options.

Une dernière méthode mise au point par Watsen et al. [120], déjà vue dans le cadre des équipements physiques dédiés, peut également être inclue dans cette catégorie. Il suggère d'afficher les menus 2D non plus dans la scène comme tout autre objet 3D, mais directement sur un périphérique léger et portable : un assistant personnel digital (PDA), le Palm (voir figure 2.27). Les techniques de contrôle d'application du monde 2D peuvent alors être réexploitées pleinement, le Palm agissant comme une mini-station de travail embarquée dans l'univers immersif.

**2.5.6.1.3 Positionnement : centré sur un équipement fixe** Le positionnement centré sur un équipement fixe permet d'exploiter le référentiel physique donné par la configuration, qui dans certain cas est très significatif. Sur un Plan de Travail Virtuel, on pourra par exemple placer le menu à une position fixe par rapport aux cadres des écrans qui constituent un référentiel fort. Cette solution [33] ne résout pas malheureusement le problème d'occlusion cité dans le cas du positionnement centré monde.

**2.5.6.1.4 Positionnement : centré sur le corps** Le placement des menus 2D par rapport à des parties du corps est une dernière approche intéressante, faisant appel au sens proprioceptif. La proprioception [11] est "le sens d'une personne de la position et



FIG. 2.27 – Utilisation d'un PDA pour contrôler les applications au sein d'un environnement virtuel immersif.

l'orientation de son corps et ses membres". Elle procure des avantages importants pour la manipulation directe (contrôle excellent de sa propre main), des indices mnémoniques physiques (retrouver des objets centrés sur son corps) et facilite les actions gestuelles (rappel des actions).

Utilisant ce principe, Mine dispose un menu au dessus de la tête de l'utilisateur, juste hors de son champ de vision [88]. Il suffit pour celui-ci d'aller le saisir et de l'abaisser d'une main, comme un store, pour pouvoir y accéder. Relâcher sa prise renvoie le menu dans sa position de repos au dessus de la tête. Mine suggère d'utiliser le corps de l'utilisateur comme référence pour placer des éléments de menus. Par exemple, changer des propriétés de la vue peut être attribué à un menu près des yeux ou de la tête de l'utilisateur. Le corps peut ainsi fournir des indices mnémoniques de position des menus en fonction de leur fonctionnalité.

Butterworth et al. [23] présentent pour leur application de modélisation 3DM une gamme d'outils virtuels, accessibles par une ceinture d'outil, placée par défaut près de la taille de l'utilisateur. Elle le suit dans ses déplacements, toujours disponible.

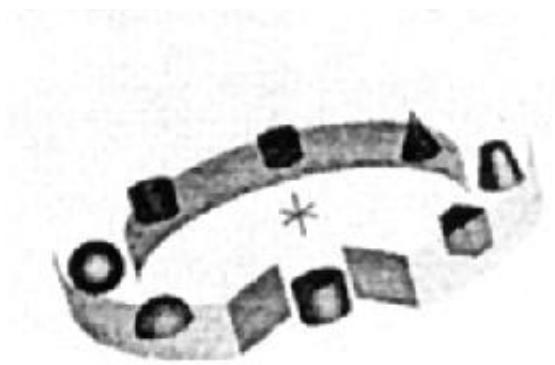


FIG. 2.28 – Menu en anneau dans le système de modélisation JDCAD [80]

#### 2.5.6.2 Menus 3D

Pour répondre aux besoins spécifiques des environnements virtuels, les concepteurs d'interface se sont penchés sur la création de nouveaux paradigmes de contrôle d'application. Une première méthode consiste à s'inspirer des techniques d'interaction des interfaces 2D et les adapter pour prendre en compte la nouvelle nature de leur environnement ou les étendre à une dimension supplémentaire. La seconde approche rompt les liens avec les interfaces des stations de travail et cherche à créer de toute part de nouveaux paradigmes purement 3D, n'ayant aucun équivalent dans le monde 2D.

**2.5.6.2.1 Paradigmes 2D adaptés ou étendus au monde 3D** Les paradigmes du monde 2D peuvent s'intégrer dans les mondes 3D en s'adaptant à ses contraintes et en évoluant pour offrir une manipulation plus confortable, ou une apparence plus adéquate.

Certains chercheurs transforment l'apparence classique des menus linéaires en redisposant les options sur une bande circulaire 3D. Le paradigme de sélection reste à une dimension tout en proposant une apparence 3D et une manipulation basée sur la rotation de la main. Liang présente dans son système JDCAD un menu où les options sont disposées sur un anneau, commandé par l'orientation d'une *bat* tenue en main [80] (voir figure 2.28). Gerold Wesche utilise un demi-anneau dont les options sont sélectionnées en les pointant à l'aide d'un rayon virtuel contraint sur une dimension [122] (voir figure 2.29).

Michael Deering propose en 1995 une adaptation 3D et hiérarchique des *pie menus* [35],



FIG. 2.29 – Menu en demi-anneau avec pointeur virtuel de Wesche [122]

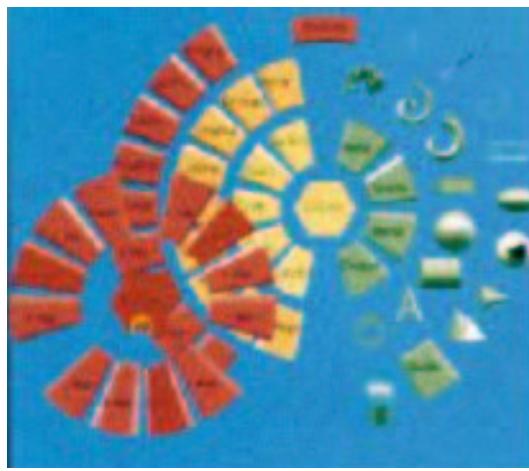


FIG. 2.30 – Menu radial dans le système Holosketch [35]

des menus radiaux qui apparaissent sur invocation autour de la position courante d'un pointeur. Des options de menu sont disposées dans des cases rangées en anneau autour d'un point central (voir figure 2.30). Dans son système, Deering fait disparaître la scène entière le temps de la manipulation du menu pour éviter les problèmes d'obstruction de vue. Le menu apparaît ensuite, et la sélection est réalisée en approchant de la case voulue l'extremité d'un *wand* tenu en main. Deering utilise la profondeur comme dimension utile pour gérer la hiérarchie. Lorsque l'option de menu donne accès à un sous menu, le disque principal recule en arrière plan et le sous-menu apparaît autour de la position courante du *wand*. Pour faire revenir le menu principal, l'utilisateur doit éloigner le *wand* du plan du sous-menu.

Chris Shaw et Mark Green développent en 1997 un système de contrôle pour une application de CAD, en proposant également une extension du concept des menus radiaux

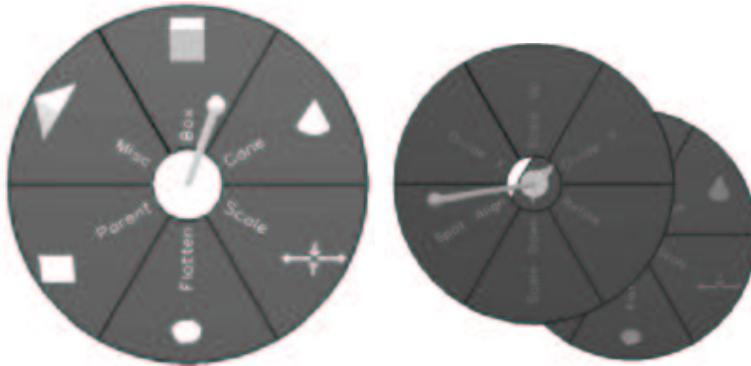


FIG. 2.31 – Le sundial menu de Chris Shaw et Mark Green

à un univers à trois dimensions. Le menu, baptisé *Sundial Menu* [107], est manipulé à l'aide d'une *bat*, un bouton monté sur un capteur de position et d'orientation. Le menu se présente graphiquement comme un disque découpé en plusieurs secteurs (représentant des choix de commandes) et faisant face à l'utilisateur. Une tige virtuelle s'articule autour du centre du disque. Son extrémité libre est contrôlée par les orientations de la *bat*. Chaque fois que le menu apparaît, la tige est réalignée avec la direction de regard. Il suffit pour sélectionner un objet du menu, de placer la tige dans le bon secteur. Une zone centrale est réservée pour l'annulation du menu sans sélection. Le *sundial menu* supporte également un système hiérarchique pour permettre l'accès à un vaste nombre d'options (voir figure 2.31).

**2.5.6.2.2 Paradigmes 3D** Les environnements virtuels offrent des possibilités d'interaction nouvelles. Une dernière voie s'intéresse à la création de nouveaux paradigmes spécifiquement conçus pour ces environnements. Plutôt que de chercher à recréer des mécanismes présents dans les interfaces 2D, ils essaient de développer de nouvelles façons d'interagir, plus directe et plus naturelle.

En 2001 Bowman et al. [18] présentent un nouveau menu baptisé TULIP (*Three-Up, Labels In Palm*), dont le principe repose sur le pincement des doigts de la main. Des *Pinch Gloves*<sup>TM</sup> sont utilisés comme gants pour détecter le pincement des doigts, tandis qu'un visio-casque permet d'afficher des options de menus comme des étiquettes rectangulaires virtuelles prolongeant les doigts, comme le montre la figure 2.32. Lorsque le pouce et un des quatre doigts opposés se touchent, l'option correspondante est sélectionnée. Pour étendre ce faible nombre d'options accessibles à un nombre quelconque d'options de menu,

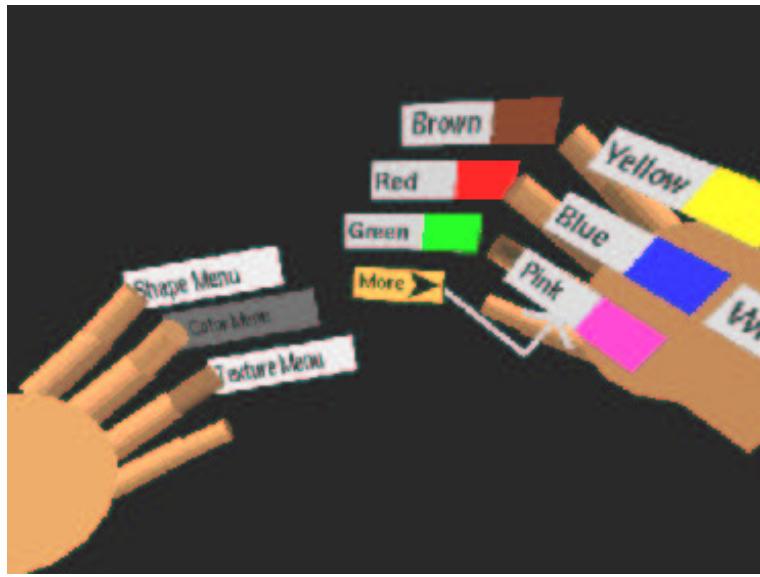


FIG. 2.32 – Menu TULIP de Doug Bowman

le pincement avec l'auriculaire est réservé à une action spéciale. Il sert à changer les options affichées sur les trois autres doigts. Lorsque l'auriculaire et le pouce se touchent, les trois options sur l'annulaire, le majeur et l'index sont remplacées par les trois options suivantes du menu. Sur la paume de la main, trois étiquettes supplémentaires sont affichées. Elles correspondent aux trois options qui remplaceront celles actuellement disponibles sur les doigts. Un aspect intéressant du paradigme est qu'il combine une interaction graphique et gestuelle. Une hypothèse émise par Bowman pour l'utilisation du menu TULIP est que des utilisateurs expérimentés puissent commencer à appeler des fonctions sans recourir au retour visuel du menu graphique. La technique d'interaction, commune à l'approche graphique et gestuelle, permet au sein d'un même paradigme de profiter des avantages de simplicité et d'intuitivité des menus graphiques, favorisant l'apprentissage, et des avantages de rapidité et d'absence d'occlusion visuelle des menus gestuels. Bien que dans les expériences menées sur TULIP, aucun des sujets n'a tenté d'utiliser TULIP en aveugle, l'approche duale suggérée reste malgré tout prometteuse.

Un autre système proposé par Multigen [66] propose de contrôler un petit jeu d'options à l'aide de deux *Pinch Gloves<sup>TM</sup>*. Lorsque la dernière phalange d'un doigt entre en contact avec celle d'un autre, une action donnée est déclenchée. L'utilisateur peut ainsi pincer le pouce et l'index d'une même main, ou toute autre combinaison entre le pouce et un autre doigt de la même main. Il peut également mettre en contact deux doigts de mains différentes.

### 2.5.6.3 Evaluations

Les techniques de contrôle d'application en environnement virtuel précédemment décrites sont souvent développées au sein d'une application ou pour servir un projet spécifique et sont encore rarement testées en elles-mêmes de façon formelle. Des tests qualitatifs auprès d'utilisateurs permettent de recueillir des impressions et un premier jugement sur ces interfaces. La tenue de tests quantitatifs, rigoureux et donc plus longs est encore peu pratiquée pour ces interfaces innovantes.

Un certain nombre d'auteurs se sont intéressés à cet aspect et fournissent des conseils sur la mise en place de tests pour améliorer l'évaluation des futures techniques d'interaction [51, 98, 19].

Bowman évalue notamment son menu TULIP [18] individuellement, et en comparaison à d'autres menus 3D existants.

### 2.5.6.4 Conclusion

Quelques chercheurs tentent d'homogénéiser et de faciliter le développement d'interfaces immersives 3D en proposant des *toolkits*, i.e. des composants logiciels élémentaires pour la construction de ces interfaces. Ils développent ainsi le concept de *widgets*<sup>2</sup> 3D [30, 127].

## 2.5.7 Saisie de chaînes de caractères

Les configurations de réalité virtuelle, de par leur caractère immersif changent profondément notre façon d'interagir avec les applications. Les périphériques d'interaction sont différents de ceux des stations de travail, et certaines opérations triviales sur stations doivent être reconsidérées partiellement ou entièrement dans les environnements virtuels. Une opération aussi simple que la saisie de caractères demande une attention particulière dans un univers 3D immersif. La pauvreté en nombre de signaux d'entrée (nombre de boutons ou touches) des équipements classiques d'interaction, vus dans la première section

---

<sup>2</sup>Un widget peut être défini comme la somme d'un élément graphique et d'un comportement.

de ce chapitre, crée une différence fondamentale entre ces deux environnements. Pour la plupart des configurations de réalité virtuelle, il n'est pas prévu d'emplacement pour poser un clavier. Contrairement aux stations de travail, qui supposent l'utilisateur assis face à un moniteur, la grande majorité des configurations de réalité virtuelle à base de grands écrans fonctionne sur l'idée d'un ou plusieurs utilisateurs debout et pouvant évoluer dans un certain espace, autour de modèles stéréoscopiques affichés en trois dimensions. Dans ce contexte différent, il est souvent peu souhaitable de placer un clavier dans un endroit fixe de l'espace. Le clavier et son support occulteraient une partie de l'univers virtuel en permanence, pour une utilisation qui reste ponctuelle, et contraindraient l'utilisateur à revenir à une position donnée de l'espace chaque fois qu'il veut entrer des chaînes de caractères.

Il n'est pas non plus envisageable de demander à l'utilisateur de quitter l'environnement pour aller chercher un clavier, à chaque besoin de saisie. Il est souvent relié par de nombreux fils ou câbles d'alimentation à des équipements immobiles et se déséquiper à chaque fois n'est pas une solution viable.

Une solution alternative consiste à employer un clavier miniature attaché au bras de l'utilisateur, pour éviter de quitter l'environnement et proposer un positionnement du clavier non envahissant, au repos et en utilisation. Cette solution demande cependant à l'utilisateur de porter un équipement supplémentaire en permanence pour une tâche ponctuelle. Dans un type d'environnement où l'utilisateur travaille souvent debout et les bras levés, la multiplication des équipements portés n'est pas nécessairement une approche souhaitable.

Dans cette optique, il peut être intéressant de conserver les périphériques d'interaction déjà présents dans ces configurations pour la manipulation ou le contrôle des fonctions de l'application. Des solutions ont été développées s'appuyant par exemple sur les gants de données.

Evans propose une méthode de saisie de texte à deux mains baptisée *VType* [39], similaire à la saisie au clavier. L'utilisateur tape dans le vide du bout des doigts, comme si un clavier s'y trouvait (voir figure 2.33). Les gants fournissent des données sur la flexion des doigts, cependant un traitement supplémentaire est nécessaire pour identifier la touche pressée. Pour un clavier QWERTY, plusieurs lettres correspondent notamment à un même doigt fléchi. Les poignets sont immobiles, seuls les mouvements des doigts sont envoyés



FIG. 2.33 – Le système de saisie de texte *VType*

à un moteur de reconstruction qui utilise dictionnaire et règles grammaticales pour déterminer quelles sont les lettres effectivement pressées et quels mots elles forment. Les problèmes de corrélation dans les mouvement des doigts (flexion d'un doigt entraînant physiologiquement la flexion du doigt voisin) ou de flexion plus ou moins importante des doigts pendant la frappe selon les utilisateurs sont pris en compte par des traitements spécifiques. Un signal sonore est également émis à chaque frappe correctement enregistrée. *VType* permet de fournir un substitut à différents types de clavier (QWERTY, Dvorak, Maltron, ordre alphabétique, etc.). Il ne permet pas cependant de corriger la frappe en cas de saisie erronée.

Bowman présente en 2001 un paradigme de saisie de texte à l'aide de *pinch gloves* [16] se rapprochant de la saisie de texte au clavier mais substituant la frappe des touches par des pincements de doigts. L'éloignement des bras contrôle le choix de la ligne de touches sur un clavier virtuel, affiché devant l'utilisateur. Le choix des doigts pincés par l'une ou l'autre main détermine quelle lettre est sélectionnée sur cette ligne (voir figure 2.34).

Au delà des gants de données, de nouveaux équipements miniaturisés et peu encombrants peuvent également tenter de capturer les mouvements des doigts. Fukumoto et Tonomura [49] développent un système complet sans fil de saisie de texte à une seule main (voir figure 2.35). Le système repose sur des anneaux passés à chaque doigt et équipés d'accéléromètres. Ces capteurs détectent l'impact des doigts de la main sur une surface dure ou souple. L'écriture est composée de frappes simultanées d'un ou plusieurs doigts

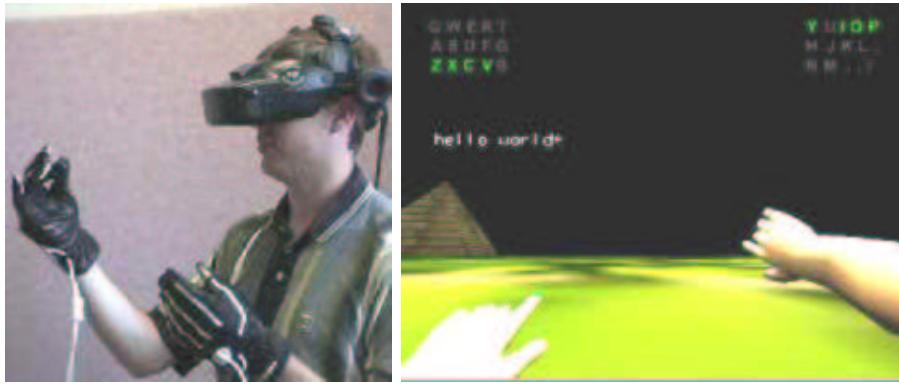


FIG. 2.34 – Saisie de texte à l'aide de *pinch gloves* par Bowman



FIG. 2.35 – Le système de saisie de texte FingeRing

pour former une gamme de symboles variant de 27 à 52 selon l'expérience des utilisateurs.

S'appuyant sur un autre périphérique physique, une palette ou tablette, Poupyrev [97] présente un bloc-notes virtuel. A l'aide d'un stylo, l'utilisateur qui porte un visio-casque écrit sur la tablette. Le texte s'imprime alors sur l'avatar virtuel de la tablette au fur et à mesure de l'écriture (voir figure 2.36).

Dans le cas où l'utilisateur n'est pas encombré par d'autres équipements tenus en main ou des gants de données, un périphérique léger et portable comme un Palm [120] peut servir à la saisie de texte dans un environnement virtuel. Toutefois cette solution occupe les deux mains de l'utilisateur, et présente les mêmes avantages et inconvénients qu'un mini-clavier embarqué. Le passage de cet outil de contrôle à des équipements de sélection et manipulation peut être problématique.

Enfin, une approche différente suivant une autre modalité, la voix, peut-être attrayante



FIG. 2.36 – Le système de saisie de texte VType

à première vue. Cependant la saisie de chaînes de caractères est ici considérée au sens large, informatique, et doit comprendre également des caractères spéciaux comme la ponctuation, les caractères numériques (i.e. 3, +, etc.) ou des caractères spéciaux (i.e. %, \$). Le vocabulaire à saisir comprend des termes ou des mots n'appartenant pas au dictionnaire, et la prononciation lettre à lettre, peut s'avérer très pénible. La voix ne se prête pas non plus facilement aux opérations fréquentes de correction ou de modification, nécessaires pour la saisie informatique.

### 2.5.8 Conclusion

Le contrôle d'application en environnement virtuel est un vaste domaine, regroupant la gestion des menus, le déclenchement de fonctions de l'application, l'appel d'outils ou encore le changement de modes. Des approches diverses peuvent être proposées, suivant plusieurs modalités, comme la voix, le geste ou l'interface graphique. Les interfaces immersives doivent prendre en compte la rapidité d'accès des fonctions, le confort d'utilisation des paradigmes, leur temps d'apprentissage et la fatigue lors d'utilisation prolongée. Il doit tenir compte de contraintes de discréption pour ne pas obstruer la vue de l'utilisateur par des composants d'interface, ou d'encombrement dans le cas d'outils physiques à porter. Des solutions ont été développées s'inspirant de celles trouvées sur les stations de travail, tandis que d'autres cherchent au contraire à proposer de nouveaux paradigmes d'interaction, purement 3D. Il n'existe pas encore cependant, contrairement aux stations de travail, de solutions universelles et le domaine du contrôle d'application est un champ

de recherche ouvert.

## 2.6 Conclusion

Un état de l'art de l'interaction en environnement virtuel a été présenté. L'interaction sur les configurations de réalité virtuelle est rendue possible par des équipements tenus en main ou portés sur le corps. Nous avons exposé une taxonomie des périphériques d'interaction. Les périphériques tenus en main sont très souvent composés d'un capteur de position et d'orientation et se distinguent par leur prise, leur forme utile, et les signaux d'entrée continus ou discrets qu'ils peuvent contrôler. Les périphériques portés se distinguent par leur localisation sur le corps, et également les signaux d'entrée continus ou discrets qu'ils peuvent contrôler.

A l'aide de ces équipements un utilisateur immergé peut interagir avec son environnement à condition d'être doté de paradigmes d'interaction pertinents et efficaces. Trois grandes catégories de tâches sont communément reconnues pour l'interaction en environnement virtuel : la navigation, la sélection/manipulation, et le contrôle d'application.

La navigation regroupe toutes les techniques permettant à un utilisateur de se déplacer dans l'univers virtuel. Deux grandes approches existent, selon que l'utilisateur cherche à se déplacer par des moyens réels, principalement la marche ou la simulation de marche, ou par des moyens virtuels, avec différentes stratégies couvrant la désignation d'une destination, d'un trajet, l'orientation de la direction de déplacement ou de la caméra.

Les techniques de sélection et de manipulation s'intéressent à la désignation, la saisie et la manipulation en translation ou rotation de modèles 3D proches ou distants, et de taille quelconque. Si aucune de ces techniques n'apporte de solution universelle, quelle que soit la distance de sélection et de manipulation où la taille des objets impliqués, elles offrent des réponses pour l'une ou l'autre de ces exigences. Une application immersive peut dès lors, selon son contexte, trouver un paradigme adapté à ses besoins. Des solutions hybrides, combinant plusieurs de ces techniques peuvent palier à leurs défauts respectifs. De même des techniques d'interaction multi-modales combinant le geste et la voix sont également très prometteuses.

Le contrôle d'application regroupe les techniques d'interaction visant à gérer l'appel des fonctions d'une application au sens large. Elle comprend des actions comme le déclenchement de fonctions, les changements de mode, ou encore la gestion des outils. Des approches très variées existent, recouvrant plusieurs modalités distinctes. Le principe général est la sélection d'une option au sein d'une liste à une dimension ou menu. La sélection peut être verbale, gestuelle, graphique, ou être directement une sélection par l'utilisateur d'outils physiques. Les menus graphiques, qui nous intéressent tout particulièrement pour la suite des travaux, peuvent être classifiés en deux types d'approches générales, suivies par la communauté.

La première approche a été de porter les menus plats linéaires du monde 2D tels quels dans le monde 3D. Le placement de ces menus est fondamental, et différentie les techniques selon que les menus 2D implantés dans l'univers virtuel apparaissent flottants dans le monde, centrés sur une partie du corps, sur un équipement mobile ou un équipement fixe.

La seconde approche, plus évoluée, s'attèle à la création de menus 3D, qu'ils s'inspirent de paradigmes 2D en les étendant ou les adaptant pour un univers 3D, ou qu'ils soient de nouveaux paradigmes 3D, totalement originaux et utilisant toutes les possibilités de la 3D.

Une dernière tâche reliée au contrôle d'application concerne la saisie de chaînes de caractères en environnement virtuel. Cette tâche apparemment triviale pour des stations de travail, doit être considérée attentivement dans un univers virtuel qui ne dispose pas de périphériques d'entrée avec beaucoup de signaux. L'utilisation d'un clavier réel n'étant généralement pas conseillée dans les configurations de réalité virtuelle, un tout petit nombre de techniques ont été développées à ce jour en se basant principalement sur les équipements physiques de l'interaction déjà présents, ou en important des interfaces matérielles dédiées, légères et portables, mais occupant les deux mains.

Le domaine de l'interaction en environnement virtuel est encore jeune, et les difficultés rencontrées pour chacune de ces tâches canoniques n'ont pas permis encore de proposer de solution universelle répondant à toutes les contraintes.

# Chapitre 3

## Contrôle d'application en environnement virtuel : le $C^3$

### 3.1 Introduction

Dans ce chapitre, nous présentons un paradigme original pour le contrôle d'application en environnement virtuel, baptisé *Command and Control Cube* ou  $C^3$ . Nous exposons dans une première partie les objectifs poursuivis pour la conception de cette interface immersive, notamment l'idée encore non explorée de fournir un mécanisme de raccourcis pour environnement virtuel. Cette recherche nous a améné à considérer une technique de raccourcis pour stations de travail, les *Marking Menus*. Le concept est ensuite étendu et optimisé pour un univers immersif. Nous décrivons le fonctionnement du  $C^3$  dans une troisième partie, suivie de son évaluation formelle auprès d'un nombre représentatif de sujets. Suite aux tests menés, nous avons décidé d'étendre le  $C^3$ , tout en conservant son paradigme d'interaction et son potentiel, à une version hiérarchique et contextuelle. Le  $C^3$  hiérarchique permet d'accéder à un nombre arbitrairement grand d'options de menu. Le  $C^3$  contextuel affiche son contenu en fonction d'un objet pointé dans la scène. Nous terminerons les développements de ce paradigme d'interaction, avec une fonctionnalité additionnelle intégrant des aides textuelles à l'exploration du  $C^3$  : les infobulles.

## 3.2 Objectifs

Comme nous l'avons vu dans le chapitre I, il n'existe pas encore de solution universelle pour le contrôle d'application en environnement virtuel. De nombreuses approches font l'objet de recherches : interfaces graphiques, gestuelles, reconnaissance vocale. Au sein des interfaces graphiques, plusieurs voies ont été explorées depuis l'intégration des menus 2D dans un univers 3D, directement ou après un travail d'adaptation, jusqu'au développement de nouveaux menus, entièrement conçus pour un univers 3D.

Sur stations 2D (moniteur, clavier et souris), une interface graphique s'est imposée comme un standard universel : l'interface WIMP. Elle contrôle l'accès aux fonctions de l'application, par trois grandes approches :

- des menus hiérarchiques, regroupant l'ensemble des fonctions de l'application.
- des menus contextuels, regroupant autour des objets ciblés par le pointeur de la souris les fonctions adjointes.
- des raccourcis, pour accéder rapidement à certaines fonctions fréquemment utilisées.

Il n'existe pas d'équivalent de cette interface populaire pour le monde 3D des environnements virtuels. Le contrôle d'application pour ces environnements est encore un domaine immature. Nous avons vu dans le chapitre état de l'art que la simple conversion d'une interface populaire dans un premier environnement n'est pas forcément le meilleur choix pour un autre environnement, dont les caractéristiques diffèrent notablement pour l'interaction.

Confrontés au problème du choix d'une interface pour dialoguer avec notre application, nous nous sommes intéressés en premier lieu à la possibilité de fournir aux utilisateurs du Plan de Travail Virtuel une première interface rapide et simple, pour un jeu réduit de commandes, correspondant aux plus usitées au sein de chaque application. Cette approche tend à recréer pour les environnements virtuels un mécanisme qui existe sur stations de travail sous le nom de raccourcis clavier. En effet dans les interfaces graphiques usuelles du monde 2D, la méthode la plus rapide pour envoyer une commande à l'application est un raccourci clavier, c'est à dire une touche ou une combinaison de touches à laquelle est associée une commande. Par exemple, la combinaison de touches CTRL-S est généralement réservée pour la sauvegarde d'un travail. Les touches du clavier peuvent être

accédées très rapidement par un utilisateur averti, et elles recouvrent en général un petit jeu de commandes, du fait de la charge cognitive des associations touche-commande. Les travaux antérieurs dans le domaine du contrôle d'application en environnement virtuel se sont concentrés sur l'implémentation de menus génériques complets et à notre connaissance aucun d'eux n'a abordé cet aspect raccourci, à savoir un menu de taille réduite mais très rapide d'accès.

Une interface graphique présente un avantage de clarté à travers un retour visuel d'information pour un utilisateur manipulant un menu. En contrepartie, elle ralentit l'utilisateur si celui-ci doit consulter ce retour visuel pour la manipulation elle-même. Les raccourcis claviers s'affranchissent du retour visuel pour n'offrir qu'une technique rapide d'accès aux fonctions. Cependant les fonctions qu'ils manipulent se retrouvent également dans des menus déroulants graphiques, disponibles à tout instant. Cette dualité permet un fonctionnement novice et expert particulièrement appréciable. Un utilisateur novice accède aux fonctions de l'application à travers une interface graphique. Un utilisateur expérimenté peut commencer à faire appel aux raccourcis dont il a mémorisé l'association touche-fonction. Il peut choisir à tout instant et pour chaque fonction l'un ou l'autre mode, selon son niveau d'expertise courant.

Dans une configuration d'environnement virtuel comme le Plan de Travail Virtuel, l'interaction entre l'utilisateur et l'application est réalisée à l'aide d'un ensemble d'outils et de périphériques portés ou tenus en main, qui ne comprennent généralement pas la souris et le clavier. Dans la présentation des équipements physiques de l'interaction du premier chapitre, nous avons vu en effet que les outils tenus en mains sont généralement construits autour d'un capteur de position et n'incluent qu'un nombre très limité de signaux d'entrée discrets comme des boutons. D'autre part, les équipements générant des signaux continus, comme les gants de données, sont sans doute moins adaptés au contrôle d'application. Postures et gestes de la main peuvent être analysés pour servir de signaux d'entrée à un système de commandes. Cependant les systèmes gestuels demandant un apprentissage certain, sont plus prompts aux erreurs de commandes et peuvent éventuellement fatiguer les doigts lors d'utilisation prolongée, ou de postures non naturelles.

Si cette pauvreté en nombre de signaux des équipements d'interaction n'est pas en soi un obstacle au développement d'un système de menu (puisque le stylus par exemple, s'apparente à une souris), elle ne permet pas de calquer directement le modèle à accès

direct des raccourcis claviers dans un environnement virtuel.

### 3.3 Les *Marking Menus*

Une solution parallèle aux raccourcis claviers et utilisant la souris existe également au sein de la métaphore du bureau : les *Marking Menus* [75, 74]. Cette technique encore peu répandue s'appuie sur une répartition spatiale du menu 2D autour du pointeur de la souris et une sélection par choix de direction. Elle intègre également un menu graphique d'apprentissage.

Les *Marking Menus* sont des menus *pop-up* radiaux. Lorsque l'utilisateur presse un bouton de la souris, un menu graphique sous forme de disque apparaît, centré sur le pointeur de la souris. Le disque est découpé en plusieurs cadrans de même ouverture. Chaque angle définit ainsi une zone de sélection. Le pointeur de la souris occupe un disque de plus petite taille, au centre de ce menu. Si l'utilisateur relâche le bouton de la souris, lorsque le pointeur est dans le petit disque central, aucune sélection n'est effectuée. Par contre s'il déplace d'abord le pointeur dans une direction de son choix, quitte le disque central, entre dans un des cadrans du menu et relâche le bouton, alors il sélectionne l'option associée à ce cadran.

Contrairement aux menus déroulants traditionnels, ici c'est la direction du mouvement de sélection qui importe et non sa profondeur. Cette approche permet des sélections plus rapides comme le montre les travaux de Callahan et al.[25]. Il est en effet plus facile de mémoriser une direction qu'une distance, et donc de se souvenir de la position des options dans le menu.

Cette facilité cognitive permet dans un second temps, pour un utilisateur averti, de réaliser des sélections "en aveugle", c'est à dire sans avoir recours au retour visuel du menu graphique. Les *Marking Menus* proposent un mécanisme à deux vitesses pour permettre un apprentissage et une transition fluide entre la méthode de sélection classique d'un menu graphique *pop-up* et la sélection en aveugle, basée sur la direction du geste de sélection. Un délai d'affichage est ajouté au menu radial. Ce délai de courte durée - de l'ordre du tiers de seconde - s'écoule naturellement pour un utilisateur débutant ou ignorant la position de l'option désirée à l'intérieur du menu. Un utilisateur chevronné ou connaissant

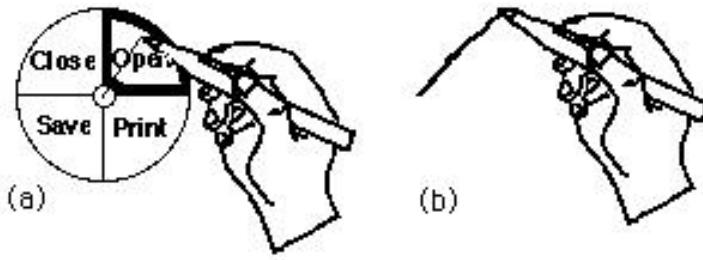


FIG. 3.1 – Sélection dans un marking menu en mode novice avec le retour visuel (a) ou en mode expert en aveugle (b)

parfaitement la position de l'option dans le menu réalisera son geste avant la fin de ce délai et relâchera le bouton de la souris avant que le menu ne puisse apparaître. Il sélectionnera ainsi "en aveugle" et très rapidement l'option de menu (voir figure 3.1).

Dans un menu déroulant sous forme de liste, l'utilisateur doit mémoriser la longueur de son geste pour chacune des options qu'il choisit, s'il veut s'affranchir de regarder le pointeur et le menu graphique lors de la sélection. L'attention requise pour se déplacer dans ce type de menu constitue un obstacle qui interdit la sélection en aveugle. Les *Marking Menus* apportent une solution élégante au concept de raccourci, en reprenant la simplicité intuitive des menus graphiques et en l'étendant en une méthode gestuelle rapide sans briser ou changer le paradigme d'interaction.

Cette interface présente des caractéristiques intéressantes pour notre recherche d'un système de raccourci pour environnement virtuel. Elle se fonde tout d'abord sur une division en cadrants d'un espace à deux dimensions, l'écran. Le concept est naturellement extensible à un univers à trois dimensions, en suivant le même processus de division. De plus, elle utilise des gestes de sélection simples : des mouvements droits, continus, dans des directions fixes du plan. Ces gestes trouvent eux aussi une correspondance immédiate dans un univers à trois dimensions. Enfin, elle ne requiert pas une précision excessive, puisqu'elle privilégie la direction à une distance de déplacement.

Le paradigme d'interaction que nous avons appelé *Command & Control Cube* [54, 55] (abrégé en CCC ou  $C^3$ . Littéralement : C cube) a pour vocation première de remplir ce rôle de raccourci vers les fonctions usuelles d'une application. Il s'appuie sur une extension des *Marking Menus* aux environnements virtuels et plus particulièrement au Plan de Travail

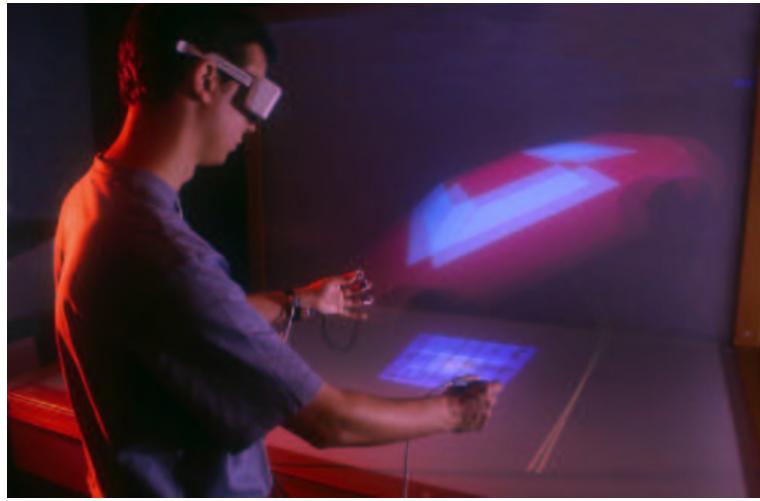


FIG. 3.2 – Utilisateur appelant le Command and Control Cube ( $C^3$ ) à l'intérieur d'une application

Virtuel.

## 3.4 Le *Command and Control Cube*

### 3.4.1 Division de l'espace sous forme cubique

Le *Command and Control Cube* ou  $C^3$  (cf. figure 3.2) est construit comme une extension du concept des *Marking Menus* [75, 74] à un univers à trois dimensions. Dans les *Marking Menus*, la zone de sélection du menu, c'est à dire l'espace autour du pointeur de la souris est divisée en cadrans identiques, délimitant les différentes portions d'un disque. Pour étendre ce concept en trois dimensions, il faut décider du nombre d'options de menu souhaitées et diviser l'espace 3D en portions homogènes. Selon le nombre considéré, par exemple une dizaine, la division de l'espace environnant en unités élémentaires identiques peut être perturbante et difficile à concevoir. Aussi, une division plus simple a été retenue.

Une première approche consiste à décomposer l'espace autour du pointeur en un ensemble de directions simples (haut, bas, gauche, droite, avant, arrière). Les options sont alors disposées de manière intuitive dans l'espace, et facilement atteignables. Cependant, le nombre d'options disponibles est alors limité au chiffre maximum de 6. Cela reste assez

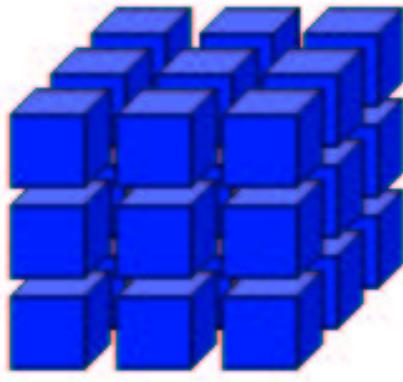


FIG. 3.3 – Espace divisé en 3x3x3 petits cubes

faible, même pour un système de raccourcis.

Une seconde approche a donc été envisagée, s'appuyant sur cette première idée pour organiser la division de l'espace en un schéma simple à appréhender et à retenir : une forme cubique.

L'espace autour du pointeur est vu comme un cube dont il est le centre. Ce grand cube est divisé selon chacune de ses dimensions en trois petits cubes pour un total de  $3 \times 3 \times 3 = 27$  petits cubes ou *cases* comme le montre la figure 3.3. Le pointeur débute dans le petit cube central de cette structure.

### 3.4.2 Le dispositif physique de commande

Pour actionner le pointeur, le *C<sup>3</sup>* nécessite un périphérique faisant office de souris 3D. Un simple bouton, dont la position spatiale est capturée par un capteur, est requis pour la manipulation. Le stylus du Plan de Travail Virtuel, le système de *Pinch Glove* développé par Fakespace [40] ou la *Ringmouse* [89] (dont la production est actuellement arrêtée) sont à ce titre des équipements valables.

Le stylus est essentiellement utilisé sur le Plan de Travail Virtuel par la main dominante pour manipuler et interagir avec les éléments de la scène, et d'une manière générale, pour réaliser la tâche principale. L'attention de l'utilisateur pendant son travail est focalisée sur l'outil tenu en main et sur la tâche en cours. Dans ce cadre il n'apparaît pas souhaitable de



FIG. 3.4 – Notre dispositif : des boutons de souris montés sur bagues et un capteur de position

devoir remplacer la fonction courante du stylus pour invoquer des commandes. Changer le paradigme d'interaction et la fonction du point focal de l'attention de l'utilisateur ne peut que le détourner de la tâche en cours et perturber le rythme de son travail, ce qui est l'objectif inverse d'un système de raccourci.

Par conséquent il est proposé que la sélection avec le  $C^3$  s'effectue avec la main non-dominante. Pendant que la main dominante travaille avec le stylus ou tout autre périphérique, la main non dominante permet de changer de mode, activer des options, changer l'outil de la main dominante, etc. sans stopper le fil des actions de la main dominante, qui est prioritaire.

Pour des raisons de simplicité, de légèreté et de coût, nous n'avons pas utilisé de système commercial pour manipuler le  $C^3$  mais en avons construit un, à partir d'une souris standard et d'un capteur de position (cf. figure 3.4). Les boutons de la souris servent de "boutons à pincer" et sont placés au sommet de trois doigts de la main à l'aide de bagues en tissus. Ils peuvent être pressés avec le pouce. Le capteur est placé sur le poignet de l'utilisateur. Un seul des trois boutons est effectivement utilisé pour les besoins du  $C^3$ .

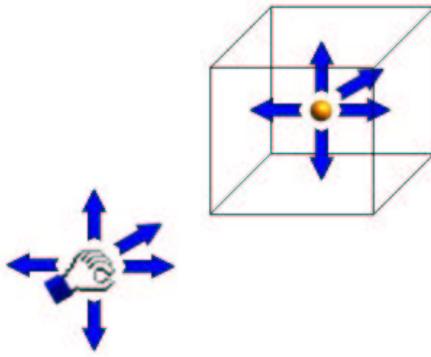


FIG. 3.5 – Les mouvements de la main sont reproduits sur la sphère-pointeur

### 3.4.3 Principe de fonctionnement

En position de repos, le menu  $C^3$  est invisible et ne gène pas la vue. Lorsque l'utilisateur presse son pouce contre son index, il pince le bouton et déclenche l'apparition du  $C^3$ , c'est à dire la forme cubique, à une distance fixe en avant de sa main. Le capteur sert à calculer cette position. Afficher le menu, relativement à la main, permet de laisser à l'utilisateur le choix du meilleur endroit où faire apparaître le menu. Le  $C^3$  reste à cette position fixe tant que le bouton est maintenu enfoncé.

Un pointeur prenant la forme d'une sphère apparaît également au centre du  $C^3$ , dans la case centrale. Sa couleur est choisie pour contraster avec celle des cubes (ici, une sphère jaune pour des cubes bleus). Tant que l'utilisateur maintient le bouton enfoncé, les mouvements du pointeur reproduisent fidèlement les mouvements de la main avec une correspondance un pour un (cf. figure 3.5). En déplaçant sa main, l'utilisateur peut placer le pointeur sphérique dans n'importe laquelle des 27 cases du menu  $C^3$ . Toute case s'illumine lorsqu'elle est pénétrée par la sphère et s'éteint lorsque celle-ci la quitte. A chaque case du  $C^3$  est associée une fonction de l'application.

Afin de ne pas induire de contraintes de précision sur l'amplitude des mouvements de sélection, il est décidé que le pointeur ne puisse pas quitter le volume cubique du  $C^3$ , même si les mouvements de la main sont plus larges. La correspondance des mouvements de la main et du pointeur est conservée tant que la sphère évolue à l'intérieur du volume du  $C^3$ . Pour des mouvements de la main au delà de cette frontière, la sphère est stoppée par l'enveloppe du  $C^3$ , positionnée à l'intersection de cette enveloppe avec le vecteur formé par la position de départ de la main et sa position courante (cf. figure 3.6). Ainsi, l'utilisateur

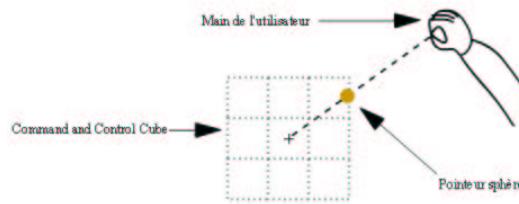


FIG. 3.6 – Le pointeur sphère est arrêté par l'enveloppe du  $C^3$

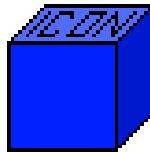


FIG. 3.7 – Une icône est placée en texture sur le sommet des petits cubes

peut librement effectuer des mouvements de sélection larges, le pointeur sphère reste contraint à l'intérieur de l'espace de sélection du  $C^3$ .

#### 3.4.4 Considérations de visibilité

Comme pour un menu 2D, il est nécessaire de pouvoir identifier les options associées aux cases d'un simple coup d'œil. A cet effet, une icône est placée sur la face supérieure de chaque petit cube élémentaire (cf. figure 3.7) sous forme d'une texture. La texture contient un pictogramme. Un texte court pourrait également être représenté, mais des tests ont montré que les résolutions des écrans dans les configurations à base de projection sur grands écrans restent souvent insuffisantes pour assurer une lisibilité satisfaisante pour la plupart des mots.

La face supérieure de chaque petit cube est choisie pour afficher cette texture. Leur visibilité est assurée par le positionnement du  $C^3$  à hauteur de la main, donc sous le regard de l'utilisateur en situation normale.

Une première difficulté au niveau de la visibilité apparaît lorsque les trois étages de cubes sont affichés en même temps, comme le montre la figure 3.8. Les étages supérieurs du  $C^3$  cachent les étages inférieurs à la vue. En conséquence il a été décidé d'afficher un seul étage de cases à chaque instant, celui correspondant à la position courante du pointeur sphérique (voir figure 3.9).

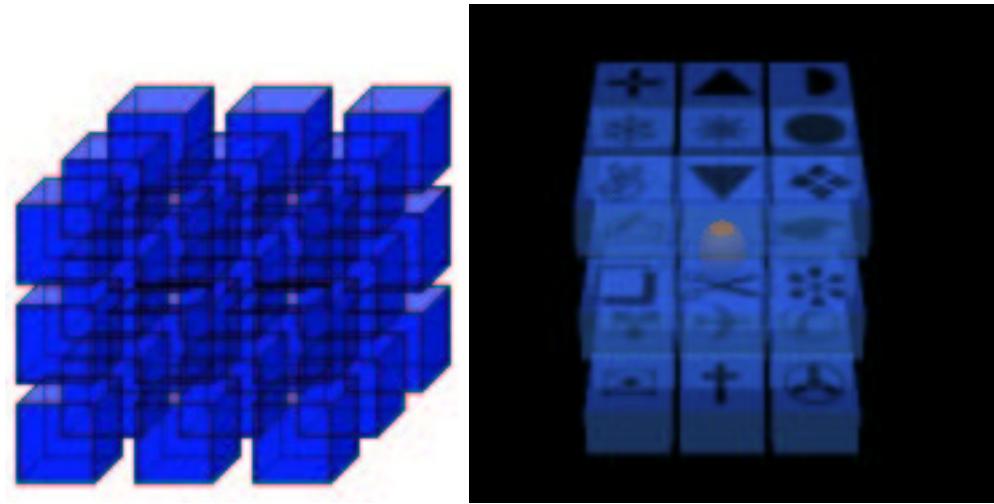


FIG. 3.8 – Les trois étages du  $C^3$  affichés ensemble posent des problèmes de visibilité des cubes et de leurs icônes

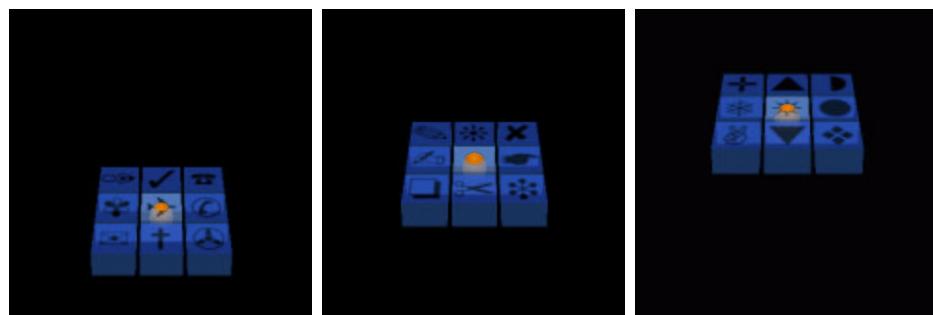


FIG. 3.9 – Un seul étage du  $C^3$  est affiché à un instant donné, selon la hauteur de la main : basse, médiane, haute

Cette méthode s'est confirmée satisfaisante et fonctionnelle dans la pratique. Cependant, après un certain temps les utilisateurs se sont montrés parfois gênés par la ressemblance des différents étages du cube et leur impossibilité de déterminer la hauteur du plateau d'un seul regard. L'absence de repère indiquant la hauteur est troublant lorsque l'utilisateur perd la notion de sa situation dans l'espace après quelques mouvements hésitants.

Pour palier à cet inconvénient nous avons implémenté et testé plusieurs solutions. Pour conserver l'apparence générale du cube, les étages absents dans la première version ont été réaffichés, mais cette fois beaucoup plus transparents qu'auparavant. Diverses valeurs de transparence ont été testées, mais il apparaît que le détail des icônes des plans supérieurs masque et perturbe malgré tout la lisibilité des icônes des plans inférieurs.

Pour alléger la représentation des cubes des étages non concernés, nous avons ensuite tenté de minimaliser la charge visuelle en n'affichant que des points au centre de ces cubes, pour indiquer leur position. La visibilité est correcte lorsque l'utilisateur a une vue rasante, horizontale sur le  $C^3$ . Les points se distinguent alors dans l'espace, mais elle reste insuffisante dans une utilisation classique où l'utilisateur a une vue plongeante sur le  $C^3$ . Le nuage de points n'est pas assez explicite et bien qu'ils ne gênent plus la vue des icônes, ils sont trop petits pour se distinguer eux mêmes sur le fond, c'est à dire sur le seul étage de cubes affiché.

Une représentation des cubes absents, de la plus opaque et visible à la plus transparente et discrète, ne permet pas de satisfaire simplement les deux critères de lisibilité des icônes importants et d'information sur la hauteur du plateau. Notre troisième version du  $C^3$  prend en compte ce constat et propose d'identifier la hauteur du plateau affiché par rapport à un repère vertical. Une simple règle verticale le long d'un bord du  $C^3$  permettrait de réaliser cet objectif. Afin de s'éloigner du visuel en plateau et renforcer la représentation mentale du cube, l'enveloppe cubique complète du  $C^3$  est affichée en transparence. Cette fois, seules les parois du cube englobant, i.e. des plans de couleur unie, sans motif, sont rendus transparents et la lisibilité des icônes n'est pas altérée. La position du plateau de cubes à l'intérieur de cette enveloppe permet à l'utilisateur de se repérer instantanément (cf. figure 3.10).

Pour différencier d'un seul coup d'œil l'étage courant à l'intérieur de la structure cubique, l'enveloppe transparente du cube englobant est également affichée (cf figure 3.10).

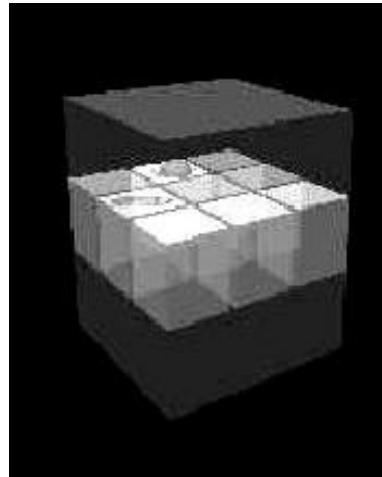


FIG. 3.10 – Le  $C^3$  avec son enveloppe transparente

### 3.4.5 Modes novice et expert, un seul paradigme

Les 27 cases sont configurables pour accueillir n'importe quelle icône et fonction de l'application, à l'exception de la case centrale, qui est réservée comme case d'annulation pour sortir du menu sans invoquer de commande. Ainsi, si l'utilisateur presse par inadvertance le bouton, il déclenchera l'option correspondant à la case où débute le pointeur sphérique, la case centrale, et ne déclenchera donc pas une commande quelconque.

La sélection d'une fonction à l'intérieur du menu est donc réalisée par la simple séquence suivante : pincer les doigts pour faire apparaître le  $C^3$  en face de sa main, maintenir le bouton et bouger la main dans une direction pour placer la sphère dans la case voulue, relâcher le bouton une fois dans cette case pour activer la fonction associée.

Le  $C^3$  propose deux modes de fonctionnement, un mode novice et un mode expert, basés sur la même technique de sélection. Dans les *Marking Menus*, un délai est utilisé pour retarder l'affichage du menu circulaire et permettre à un utilisateur expérimenté de réaliser des sélections "en aveugle", sans être perturbé par l'affichage bref du menu. Un utilisateur novice laisse naturellement s'écouler ce délai (de l'ordre du tiers de seconde) et effectue ses sélections en utilisant le retour visuel. Un utilisateur chevronné connaît par cœur la disposition des options dans le menu et peut choisir la direction de son mouvement, avant de voir le menu s'afficher.

Cette méthode a été appliquée au menu  $C^3$ . Contrairement aux attentes, des tests

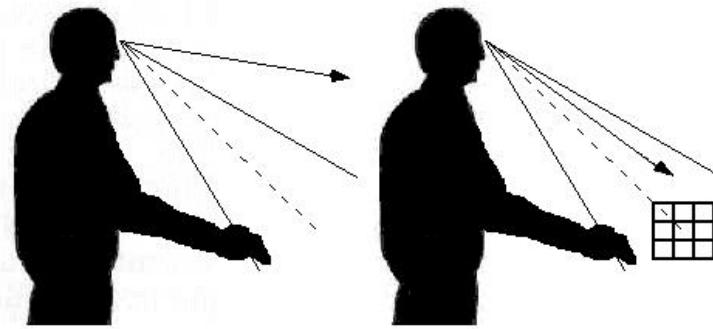


FIG. 3.11 – Affichage du  $C^3$  selon la direction de regard

informels avec plusieurs valeurs de délai n'ont pas donné un confort suffisant à la manipulation. Pour des valeurs faibles, le délai n'est pas suffisant pour couvrir le temps nécessaire au mouvement de la main. Pour des valeurs plus importantes cependant, le délai devient une gène pour l'utilisateur novice, qui ressent une attente avant l'apparition du menu et le début de chaque sélection. Le mouvement de la main dans l'espace, pour la sélection, est légèrement plus lent que le bref mouvement de la main sur la souris. Cette différence suffit à compromettre la recherche du compromis idéal entre les deux contraintes : ne pas dépasser en vitesse l'utilisateur averti, ne pas ralentir l'utilisateur novice. Une seconde approche a donc été choisie, où le  $C^3$  s'adapte "intelligemment" aux besoins de l'utilisateur.

Une observation naïve montre que l'utilisateur averti ne s'intéresse pas au retour visuel du  $C^3$  et ne détourne donc pas la tête pour le regarder. A l'opposé, l'utilisateur novice s'intéresse à l'affichage graphique et baisse spontanément la tête vers l'emplacement connu où doit apparaître le  $C^3$ , même lorsque celui-ci n'est pas encore là du fait du délai. Une solution élégante consiste donc à analyser la direction de regard de l'utilisateur pour deviner son intention.

En utilisant les données venant du capteur positionné sur la tête en plus de celles de la main, il est possible de déterminer la direction de regard de l'utilisateur et de vérifier si elle entre dans un cône de sommet la tête et d'axe tête- $C^3$ . Ainsi, lorsque l'utilisateur est concentré sur une tâche en cours et ne se préoccupe pas du menu, celui-ci n'apparaît pas, et lorsqu'il hésite ou décide qu'il a besoin d'une confirmation visuelle, il déplace son regard spontanément et tout naturellement vers l'endroit où celui-ci apparaît (voir figure 3.11).

La flexibilité du système est totale. Un utilisateur peut commencer son mouvement en

aveugle, confiant sur sa connaissance de la disposition des options dans le  $C^3$  mais hésiter soudain à relâcher le bouton. Il lui suffit de tourner la tête vers l'emplacement du  $C^3$  le temps de le faire apparaître pour confirmer la justesse de sa sélection ou au contraire corriger son mouvement. Le système ne s'adapte plus au niveau d'expertise de l'utilisateur mais plus souplement à son intention ponctuelle.

### 3.4.6 Fondements du mode aveugle

La sélection en aveugle, qui doit être intuitive et rapide, est rendue possible par l'exploitation de plusieurs indices de positionnement.

Elle s'appuie d'abord sur la mémorisation d'une direction plutôt qu'une distance selon une direction donnée (comme c'est le cas pour les menus déroulants). Il est plus facile de se souvenir d'une direction que d'une distance. La notion de distance nécessite mesure et mémorisation, alors que la notion d'angle, bien que quantifiable est remplacée par l'idée intuitive de direction. Les travaux de Callahan et al. [25] ont montré la plus grande efficacité de cette méthode en terme de vitesse. L'éloignement faible et constant entre le point de départ et chaque item de menu contribue à la rapidité de sélection.

Ensuite elle s'appuie sur une organisation simple de l'espace en cases formant une structure cubique. La représentation cubique fournit un cadre mental simple pour visualiser l'espace autour du pointeur et la direction du trajet à réaliser. Toutes les cases de destination sont adjacentes par une face, un bord ou un coin à la case centrale, position de départ. Chaque mouvement se résume à dessiner dans le vide un trait aligné sur les axes du cube ou suivant une de ses diagonales.

La représentation mentale de ces axes de déplacement est renforcée par leur adéquation à deux repères forts pour l'utilisateur. D'une part son propre corps, car pour un utilisateur faisant face au Plan de Travail Virtuel l'alignement du cube coïncide avec des directions simples relatives à son corps : haut, bas, droite, gauche, avant, arrière. Tout mouvement de sélection suit une combinaison de ces directions simples, et s'appuie sur le sens proprioceptif de l'utilisateur. D'autre part les axes du Plan de Travail Virtuel lui-même, qui sont en permanence dans le champ de vision de l'utilisateur, rappellent les axes du  $C^3$ , sur lesquels il est aligné.

### 3.5 Evaluations expérimentales et analyses

L'objectif principal de l'étude [52] est de mener une évaluation du  $C^3$ , et mesurer son apport pour la tâche standard de sélections répétées d'options à l'intérieur d'un menu, pour le Plan de Travail Virtuel. Une expérience est menée pour collecter et analyser formellement les performances de sujets utilisant le  $C^3$ . La tâche demandée est celle de sélection d'options dans un menu 3D.

L'étude s'efforce pendant cette tâche de rassembler et mesurer tous les paramètres pertinents concernant le cube. Du fait de la spécificité spatiale du procédé de sélection du  $C^3$ , nous nous intéressons notamment à l'effet de la position des options à l'intérieur du cube sur les performances (vitesse et précision). D'autre part, les performances des utilisateurs dans le mode aveugle requièrent toute notre attention. Il est en effet très important pour le mécanisme d'apprentissage et le paradigme du  $C^3$ , que ceux-ci présentent une aisance suffisante pour permettre une manipulation naturelle avec la main non dominante [67] et en l'absence de tout retour visuel.

La difficulté potentielle de cette tâche de sélection pourrait cependant être allégée par l'ajout de retour d'informations sur des canaux sensoriels différents de la vue. L'apparition furtive d'un menu graphique à la périphérie du champ de vision est indésirable, cependant l'utilisation d'une modalité sensorielle différente peut être une méthode efficace et discrète, à évaluer, pour transmettre une information de position spatiale sans détourner l'œil de l'utilisateur. Dans ce cadre, l'étude s'est portée sur l'ajout de deux autres modalités.

Deux des quatre conditions sont les modes standards de fonctionnement du  $C^3$ , i.e. le mode d'apprentissage (avec retour visuel), et le mode expert (sans retour visuel, c'est à dire une manipulation "en aveugle"). Deux conditions additionnelles sont construites en associant au mode expert des retours sur des canaux sensoriels différents : mode expert augmenté d'un signal sonore (sans retour visuel, avec retour sonore), mode expert augmenté d'un signal tactile (sans retour visuel, avec retour tactile). Durant les premiers tests conduits, l'ensemble des combinaisons des trois conditions visuelle, tactile et sonore ont été considérées (sonore et tactile, visuel et tactile, sonore et visuel, sonore visuel et tactile). Malheureusement la durée totale de l'expérience devient alors très conséquente (une heure et demi à deux heures par sujet). L'attention des sujets déclinant très sensiblement au delà de la première heure de test, il a été décidé de ne pas conserver ces

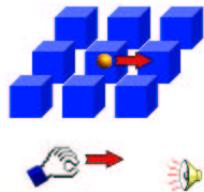


FIG. 3.12 – Bip sonore au passage d'une frontière entre cases

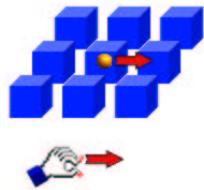


FIG. 3.13 – Vibration tactile au passage d'une frontière entre cases

conditions combinées pour le test final et d'évaluer pour cette expérience les résultats des différentes modalités, prises isolément.

Dans le mode expert augmenté d'un signal sonore, un court bip sonore est émis chaque fois que le pointeur croise la frontière entre deux cases (voir figure 3.12). Le mode expert augmenté d'un signal tactile fonctionne de la même manière en émettant une courte vibration sur le pouce et l'index de l'utilisateur au passage d'une frontière (figure 3.13). Un gant CyberTouch<sup>TM</sup> (un gant équipé de vibrateurs sur la dernière phalange de chaque doigt et dans la paume) est utilisé pour produire les vibrations. Puisque nous voulons tester le *C*<sup>3</sup> dans son mode de fonctionnement standard, i.e. sélection par la main non-dominante, le CyberTouch<sup>TM</sup> utilisé est un gant gauche pour des utilisateurs droitiers.

A noter que pour les modes experts augmentés d'un signal sonore ou tactile, des signaux d'intensité constante par case ont tout d'abord été testés. Chaque case posséde une intensité différente, mais la vibration est permanente. Cette solution a été rejetée pour le désagrément que cause un bourdonnement permanent sonore ou tactile.

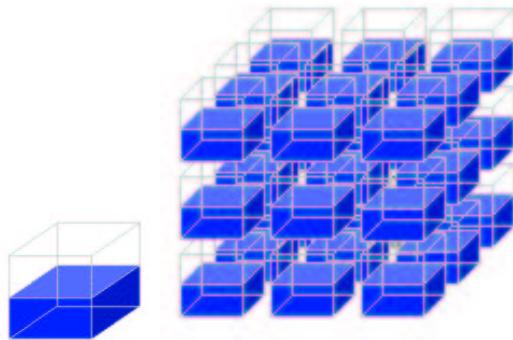


FIG. 3.14 – Cases à demi “remplies”

### 3.5.1 La version du $C^3$

Les tests ont été menés sur la toute première version du  $C^3$ , qui ne comprend pas encore l'enveloppe englobante transparente. De plus dans cette toute première version, chaque bloc est représenté avec une demi-hauteur et occupe la portion inférieure de la case cubique (cf. figure 3.14). Ce design initial s'inspire de l'idée que le pointeur pouvait être difficilement visible à travers des cubes entiers, même translucides. Un demi-cube laisse la moitié supérieure du pointeur sphérique apparaître. Les motivations de cette approche cependant se sont révélées par la suite non fondées et ont été abandonnées. Elle éloigne indésirablement du modèle conceptuel de la structure cubique en imposant une image en paliers, non isotrope et donc plus difficile pour la manipulation en aveugle.

### 3.5.2 Les Sujets

La population de testeurs est composée de 23 sujets (18 hommes et 5 femmes). Seulement quatre d'entre eux sont déjà familiers du Plan de Travail Virtuel. Aucun n'a d'expérience préalable avec le menu  $C^3$ . Les sujets sont tous droitiers.

### 3.5.3 La Tâche

La tâche consiste à sélectionner séquentiellement chacune des 26 cases du  $C^3$  dans un ordre de passage aléatoire.

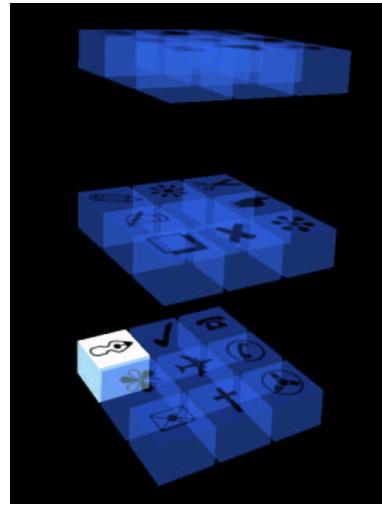


FIG. 3.15 – La cible désignant la case à sélectionner à chaque étape du test

La principale difficulté est de décider de quelle manière informer les utilisateurs de la case à sélectionner, sans leur demander au préalable d'apprendre par cœur un jeu de fonctions et leurs positions dans le cube. L'objectif est de séparer le travail cognitif consistant à reconnaître la case demandée de l'aspect manipulation qui est étudié ici. Il est donc impossible d'utiliser le nom d'une fonction pour désigner une case.

Chaque case étant une combinaison simple de directions spatiales (haut, bas, avant, arrière, gauche, droite) il pourrait être envisagé de décrire oralement la case à sélectionner. Cependant cette solution est rejetée car elle peut favoriser une stratégie de déplacement du pointeur dans le  $C^3$  (mouvement décomposé selon les trois axes) aux dépends des autres possibles (notamment réaliser des mouvements diagonaux).

Pour ne pas influencer le sujet, il est finalement décidé de lui présenter une représentation complète du menu  $C^3$  et de ses 27 cases, et d'illuminer la case à sélectionner. Les sujets savent ainsi immédiatement quelle case ils doivent sélectionner, sans qu'aucune présomption de mouvement ne soit induite. Cette représentation est un objet graphique supplémentaire dans la scène et n'est pas le menu  $C^3$  lui-même, qui reste positionné, lorsqu'il est invoqué, devant la main de l'utilisateur (voir figure 3.15 et 3.16).

Les quatre modes sont testés dans un ordre différent pour chacun des 23 sujets de l'expérience. Pour chacune des conditions expérimentales l'utilisateur porte un gant CyberTouch, le dispositif nécessaire au  $C^3$  (bouton sur le doigt et capteur), et les lunettes stéréoscopiques. Les 27 cases sont affichées devant les yeux de l'utilisateur. La case cou-

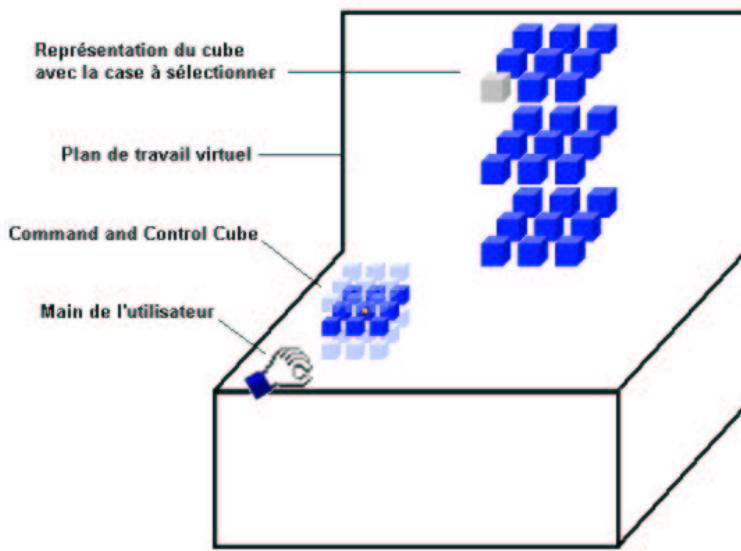


FIG. 3.16 – Le dispositif de test

rante s'illumine jusqu'à ce qu'une sélection soit réalisée. Le système informe alors l'utilisateur du résultat obtenu en illuminant en vert la case si elle a été correctement atteinte, ou illumine en rouge la case sélectionnée par erreur. Après un délai de deux secondes, la sélection d'une nouvelle case est proposée et le test continue de la même manière.

Toutes les cases sont testées deux fois en tout, à l'exception de la case centrale qui est réservée à l'action d'annulation et n'est pas testée ici.

### 3.5.4 Données Collectées

Les données enregistrées sont le flot des coordonnées des différents capteurs, la liste ordonnée des cases demandées et les sélections effectivement réalisées par l'utilisateur, et enfin les trois temps de sélection (lorsque la case s'illumine, lorsque l'utilisateur invoque le menu en pressant le bouton pour la première fois, lorsque l'utilisateur termine sa sélection en relâchant le bouton). La vitesse de sélection se déduit directement de ces données.

### 3.5.5 Résultats et discussions

L'analyse de la variance (ANOVA) est un test permettant de comparer les différences entre plusieurs moyennes (plus de 2) pour des distributions normales. Une population est divisée en plusieurs groupes, selon les conditions étudiées, et des mesures sont effectuées sur un paramètre donné pour chaque élément de cette population. Si les conditions servant à définir les groupes d'individus n'ont pas d'effet statistiquement significatif sur le paramètre de mesure, l'hypothèse zéro doit être vérifiée :

Hypothèse zéro : toutes les moyennes des échantillons (correspondants aux différents groupes étudiés, c'est à dire aussi aux différentes conditions étudiées) sont statistiquement égales. Ce test de comparaison est effectué pour un niveau de fiabilité choisi en général à 5%.

On note  $F = (\text{variation trouvée pour les moyennes des conditions}) / (\text{variation attendue pour les moyennes des conditions})$ . Ainsi, si l'hypothèse zéro est correcte, nous attendons un  $F$  proche de 1. Un  $F$  "grand" révèle un effet significatif des conditions considérées. La taille minimale de  $F$  permettant de suivre cette conclusion est liée à une facteur  $P$ , indiquant le niveau de signification. La signification statistique d'un résultat est la probabilité que la différence observée entre moyennes pour une expérience donnée n'est due qu'à la simple chance. Par conséquent, plus ce nombre  $P$  est faible, plus le résultat peut être considéré comme "fiable". Enfin, dans la suite, nous noterons pour chaque test "m" la valeur de la moyenne.

Une analyse de la variance avec mesure répétée a été réalisée sur les variables précision et vitesse pour les quatre conditions : sans retour (mode aveugle), avec retour visuel, avec retour sonore, avec retour tactile.

La vitesse est significativement affectée par la condition ( $F = 4,42$ ,  $p < 0,0068$ ). Des tests supplémentaires (HSD Tukey) indiquent que les utilisateurs réalisent rapidement la sélection dans la condition visuelle ( $m = 1,0s$ ) comparativement aux autres conditions (aveugle :  $m = 1,2s$  ; son :  $m = 1,3s$  ; tactile :  $m = 1,3s$ ). Cette supériorité du mode visuel peut s'expliquer par le fait que les utilisateurs sont tous novices. Ils ont encore besoin du support visuel pour être confiants dans leurs sélections et les réussir rapidement.

La précision est aussi significativement affectée par la condition ( $F = 10,616$ ,  $p <$

0,0001). Le pourcentage moyen de cases correctement sélectionnées est de 92,8% avec retour visuel, 87,0% dans le mode aveugle, 83,5% avec retour sonore et 84,7% avec retour tactile. Un des points d'intérêt de cette étude était de vérifier que les sélections en absence de tout retour sensoriel avec la main non-dominante étaient possibles. Considérant que les utilisateurs découvraient le menu pour la première fois, et pour certains la configuration elle-même, les résultats semblent favorables à cette hypothèse. Les résultats du mode visuel, bien que proche de 100% ne sont pas parfaits. Une explication peut être trouvée dans la contrainte de vitesse qui était demandée aux sujets du test.

Les performances plus faibles des conditions tactiles et sonores peuvent s'expliquer par la nature parfois perturbante de ces signaux. Les sujets ont remarqué qu'ils n'arrivaient pas toujours, pour les mouvement diagonaux à distinguer entre un seul bip (ou vibration) et deux ou trois bips (ou vibrations) successifs. Les sujets qui emploient une stratégie de décomposition de leur mouvements selon les trois axes n'ont pas rencontré ce problème. Dans la plupart des cas, les sujets utilisent les canaux sensoriels sonores et tactiles pour vérifier la validité de leur mouvement, plus que pour réaliser le mouvement lui-même.

La hauteur d'une case dans le  $C^3$  (position) a aussi un effet significatif ( $F = 134,058$ ,  $p < 0,0001$ ). Chaque valeur de précision représente un nombre de sélections correctes réalisées dans un plan. Pour chaque hauteur considérée, il y a 9 cases disponibles, testées deux fois chacune, soit un score maximum de 18. Le nombre de sélections correctes est plus élevé dans le plan central ( $m = 17,352$ ), puis dans le plan supérieur ( $m = 15,587$ ) et enfin dans le plan inférieur ( $m = 12,859$ ). Les mouvements à hauteur de la main sont les plus aisés. Une explication possible est le moindre effort du bras requis par les mouvements dans le plan central par rapport aux mouvements qui changent de niveau. Il faut noter également que la présence de l'écran horizontal du Plan de Travail Virtuel sous la main de l'utilisateur a pu gêner ou retenir les utilisateurs dans leur intentions de mouvement vers le bas.

L'interaction double du mode et de la position est également significative ( $F = 2,684$ ,  $p < 0,0172$ ). Cette interaction peut être interprétée dans le tableau de la figure 3.17.

Chaque ligne du tableau présente une condition différente, alors que les colonnes sont les cases des trois plans horizontaux, du plus bas au plus haut (cf. figure 3.18 pour la numérotation des cases). Les cellules contiennent la somme de toutes les réponses correctes pour les 23 utilisateurs pour une case (avec un maximum de 46 bonnes réponses). La valeur

	lower plane									middle plane								upper plane								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
blind	32	39	35	36	29	36	40	34	45	46	46	45	45	43	44	45	46	44	40	43	39	29	39	41	40	40
visual	36	42	43	42	37	41	43	43	42	44	46	45	46	46	44	46	44	42	43	43	39	43	42	44	42	42
sound	29	34	32	29	33	33	41	38	37	44	44	44	41	44	42	41	40	41	41	42	39	36	37	40	42	36
tactile	29	36	37	32	26	33	37	38	41	45	46	46	46	45	45	42	42	46	43	38	40	41	29	40	37	40

FIG. 3.17 – Nombre de réponses correctes en fonction de la position des cases (nombre vert : 100% correct ; nombre bleu : score au dessus du premier quartile ; nombre rouge : score en dessous du premier quartile ; nombre noir : scores restants)



FIG. 3.18 – Numérotation des cases

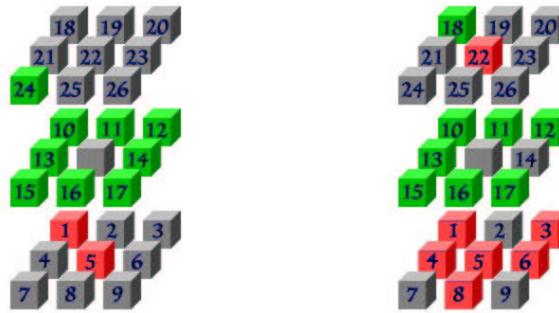


FIG. 3.19 – Résultats en mode visuel et aveugle (cases vertes : score au dessus du premier quartile, cases rouges : score en dessous du premier quartile, cases grises : scores intermédiaires)

de la médiane est de 41 bonnes réponses. Les cellules avec des résultats en dessous du premier quartile sont en rouge. Celles avec des résultats au dessus du dernier quartile sont en bleu, et celles avec un taux de 100% sont en vert.

### 3.5.6 Premières conclusions

La distribution des résultats statistiques montre quelques tendances générales (voir figure 3.19). Les performances les plus faibles sont obtenues pour des cases dans le plan inférieur. Ce résultat s'explique par la proximité de l'écran du Plan de Travail Virtuel. Dans ce plan, les mouvements qui demandent à l'utilisateur d'allonger son bras reçoivent de moins bons résultats, tandis que ceux moins éloignés du corps obtiennent de meilleurs scores. Les meilleurs résultats sont obtenus dans le plan du milieu, avec des performances légèrement moins bonnes quand la main est trop proche du corps et légèrement meilleures quand elle s'en éloigne pour une distance plus confortable. Il semble d'une manière générale que les utilisateurs réussissent mieux les mouvements demandant un minimum de flexion de leur bras.

## 3.6 Version hiérarchique

Disposant d'une solution fonctionnelle pour contrôler un petit jeu de commandes, l'étape suivante consiste à étendre cette solution à une version hiérarchique, sous forme d'arbre de menus et de sous-menus, afin de gérer un nombre potentiellement infini d'options. Notre dispositif physique de commande comprenant les trois boutons d'une souris, nous choisissons un second bouton, porté sur un autre doigt, pour la version hiérarchique du  $C^3$ . Plusieurs approches sont envisageables pour contrôler la hiérarchie.

Dans le fonctionnement hiérarchique classique des menus déroulants, la sélection est réalisée par un mouvement continu. Le pointeur suit un parcours allant de menus en sous-menus, franchissant des frontières bien définies entre les menus. Il n'y a pas de zones de transition et les menus ne se chevauchent pas. Le point de sortie du menu père est juxtaposé à l'option choisie, et le point d'entrée dans le menu fils est toujours situé en haut du menu. Il n'y a pas de symétrie. Le segment à traverser pour descendre dans le

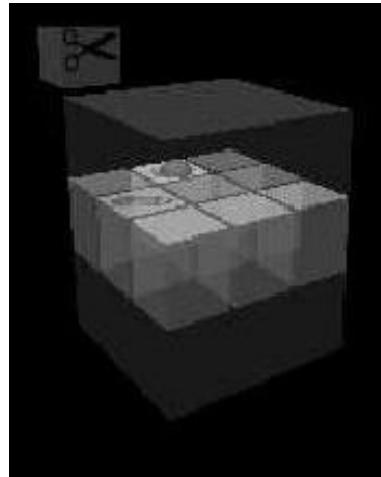
sous-menu est également souvent très étroit (de la hauteur du texte désignant l'option). Tous ces points concourent à freiner et rendre contraignante la descente et la remontée dans le système hiérarchique.

Un tel système n'est pas forcément souhaitable pour le Plan de Travail Virtuel et le  $C^3$ . Notamment, l'idée de réaliser la sélection d'un seul mouvement continu, qui sera plus large dans l'espace que le mouvement de la main sur la souris, comporte le risque de faire entrer la main de l'utilisateur en collision avec l'écran inférieur du Plan de Travail Virtuel. Cette situation peut se produire dans le cas d'une succession de sélections de sous-menus positionnés les uns en dessous des autres. D'une manière générale, l'amplitude du mouvement de sélection peut vite prendre des dimensions importantes dans le cas de nombreux sous-menus successifs.

Notre solution s'est donc portée sur l'idée de restreindre la manipulation du  $C^3$  hiérarchique à une zone délimitée de l'espace, qui soit confortable pour l'utilisateur : la position de départ du  $C^3$ , qui correspond déjà à son choix préféré. Les menus et sous-menus s'affichent tous à cette position. La géométrie (forme, placement) du  $C^3$  ne change donc pas, mais le contenu sémantique des cubes (fonctions associées et icônes) est adapté à la position du pointeur au sein de l'arbre des sous-menus.

Pour descendre dans un sous-menu associé à une case donnée, il faut sélectionner la case, c'est à dire relâcher le bouton une fois dedans, comme pour une sélection normale. S'il peut être contraignant d'avoir à décliquer pour entrer dans un sous-menu, cela est nécessaire pour distinguer l'entrée dans la case avec le pointeur, du choix de descendre dans le sous-menu. En distinguant ces deux opérations, il est possible de naviguer de cases en cases, sans que les sous-menus remplacent involontairement le contenu courant du  $C^3$  et de n'entrer dans un sous-menu que lorsque l'utilisateur l'a réellement décidé. De plus, cela maintient une cohérence d'interaction vis à vis des cases du  $C^3$ .

A l'issue de cette sélection un nouveau contenu du  $C^3$  est affiché, le menu reste visible mais cette fois la main ne presse plus le bouton. La situation ne diffère de la position de repos, avant une première invocation du menu, que par la visibilité du  $C^3$  en face de l'utilisateur et l'apparition du dernier cube sélectionné, empilé au dessus du  $C^3$  (voir figure 3.20). Dans ce cas de figure, le pointeur sphère est replacé au centre du  $C^3$ , fixe et en attente. Tant que l'utilisateur ne réappuie pas sur le bouton, celle-ci ne bouge pas. Dès qu'il appuie, il la rend à nouveau solidaire des mouvements de sa main, comme s'il la

FIG. 3.20 – Le  $C^3$  hiérarchique

saisissait, et la sélection suit toujours le même principe.

La case centrale, qui est une case d'annulation dans le premier menu rencontré, devient dans les sous-menus, une case de "remontée" vers le menu père. Si l'action de remontée est choisie, le mécanisme est similaire à la descente dans un sous-menu. Le  $C^3$  reste visible, son contenu change pour correspondre aux 26 options du menu père, le dernier cube empilé au dessus du  $C^3$  disparaît et le pointeur sphère se replace au centre du  $C^3$ , en attente d'être saisi.

Pour résumer, descendre dans la hiérarchie des sous-menus consiste à enchaîner des sélections simples, selon le même paradigme d'interaction que pour le  $C^3$  non hiérarchique, comme illustré dans la figure 3.21. Les cubes sélectionnés pour descendre dans la hiérarchie de sous-menus s'empilent successivement au dessus du  $C^3$ , montrant le chemin parcouru. Remonter dans la hiérarchie ou quitter le menu sans sélectionner est également très simple, même lorsque le pointeur est descendu profondément dans l'arbre des sous-menus. Il suffit de presser le bouton sans bouger la main. En effet, la sphère est toujours replacée en attente au centre des cubes après chaque action de montée ou de descente. Par conséquent cliquer sans bouger fait choisir la case centrale, i.e. remonter de nœuds en nœuds dans l'arbre jusqu'au nœud racine, puis sortir du menu.

Une dernière remarque peut être faite concernant une hiérarchie de menus ayant plus de 26 options à un même étage. Ce problème peut être contourné par une technique employée par Bowman dans le menu TULIP [18] : une case est réservée à l'action de

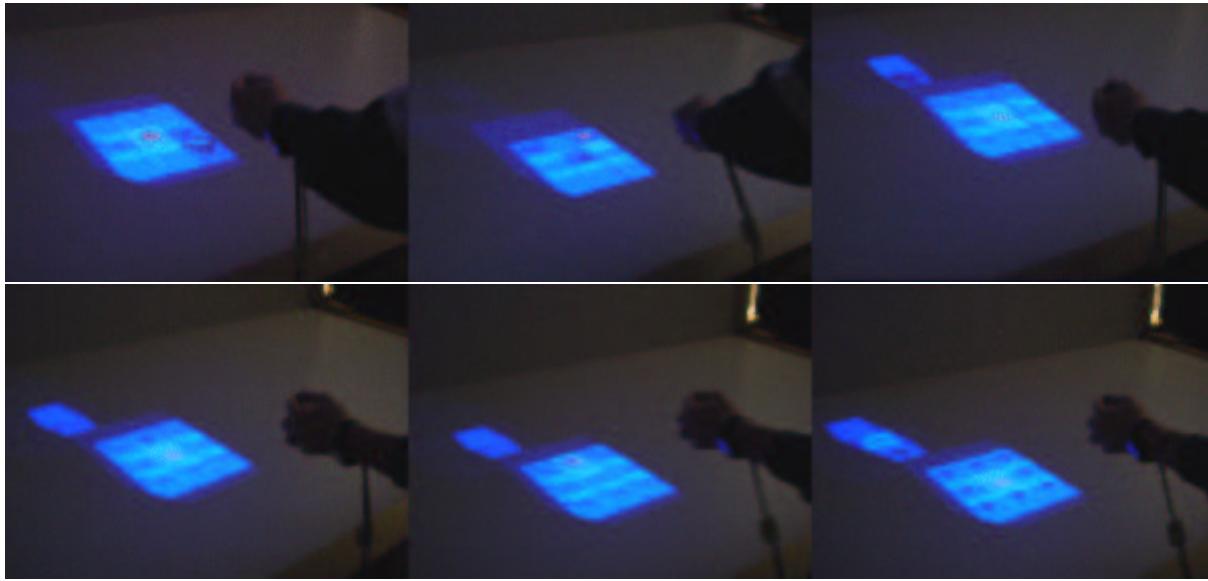


FIG. 3.21 – Descentes successives dans deux sous-menus du  $C^3$  hiérarchique

renouvellement de contenu des 25 autres. Ainsi, il est possible de faire progresser une "fenêtre" de 25 options sur une liste de taille arbitrairement grande.

## 3.7 Version contextuelle

L'interface WIMP des stations de travail 2D propose à travers ses menus déroulants, trois approches différentes et complémentaires pour accéder aux fonctions d'une application. La première est le menu déroulant fixe traditionnel, généralement positionné en haut de l'écran ou de la fenêtre courante. Il donne accès à la pyramide complète des tâches d'une application sous une forme structurée en arbre. Aux fonctions les plus communes, un raccourci clavier est associé, dont la combinaison de touches est rappelée dans le menu déroulant traditionnel lui même, à côté du nom de l'option. Cet accès rapide aux fonctions usuelles est la seconde méthode d'appel des fonctions de l'application. Elle sacrifie l'exhaustivité au profit de la rapidité. Enfin, une troisième méthode vient compléter ces deux approches : les menus contextuels. Ce sont des menus graphiques qui apparaissent sur une action de l'utilisateur -un clic souris- et dont le contenu sémantique dépend de la position dans l'espace visuel, du pointeur utilisé pour l'appel. Les menus contextuels apportent un accès à un jeu limité de commandes, cette fois ciblé sur les besoins de l'utilisateur à un instant donné.

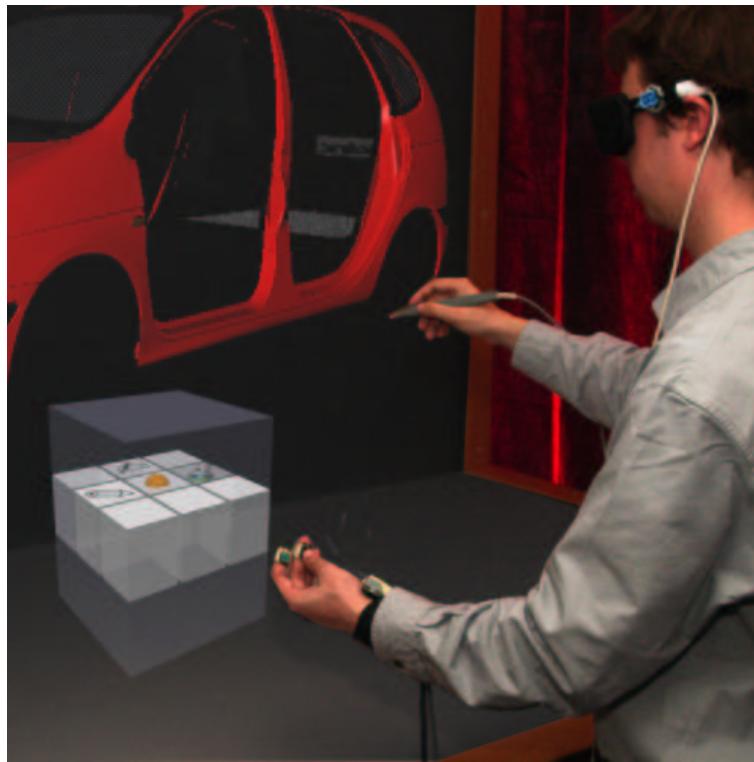


FIG. 3.22 – Utilisateur appelant le Command and Control Cube ( $C^3$ ) à l'intérieur d'une application

Les menus déroulants traditionnels trouvent leur contrepartie dans notre environnement virtuel avec le  $C^3$  hiérarchique, qui offre une structure pyramidale pour l'accès à l'ensemble des fonctions de l'application. Les raccourcis claviers, objets premier de l'étude du  $C^3$ , sont reproduits sous la forme du  $C^3$  simple. Les menus contextuels constituent la dernière pièce manquante pour reproduire un système de contrôle d'application équivalent aux stations de travail, pour les environnements virtuels. Notre dispositif physique de commande provenant d'une souris à trois boutons, nous pouvons réservé un dernier bouton pour la version contextuelle du  $C^3$ .

Il est aisément de simuler les menus contextuels en associant l'apparition d'un  $C^3$  hiérarchique ou non, à une opération de pointage avec un rayon virtuel prolongeant le stylo tenu en main dominante (voir figure 3.22). Lorsque le rayon désigne un endroit de la scène pertinent, une pression du bouton en main non-dominante fait apparaître le  $C^3$  à sa position traditionnelle, devant la main. Une fois ce contexte donné, le stylo contrôlant le rayon peut bouger dans la scène sans conséquence, le menu déjà affiché ne change pas. La sélection dans le  $C^3$  suit la procédure habituelle.

## 3.8 Info bulles

Le  $C^3$  propose dans sa version hiérarchique un accès pyramidal à l'ensemble des options d'une application. Les applications les plus complexes peuvent atteindre un nombre très conséquent de fonctions. Il peut devenir alors difficile à un utilisateur de reconnaître ou comprendre la fonction associée à une icône ou un simple mot lorsqu'elles se comptent en centaines. Une nouvelle fonctionnalité est apparue sur les environnements graphiques des stations de travail. Elle permet de guider l'utilisateur dans sa découverte des menus ou dans ses recherches de fonctions à l'intérieur des menus linéaires. Un court texte descriptif est associé à chacune des options d'un menu, et ces informations sont affichées près de l'option dans une bulle graphique (à la manière d'une bulle de texte de bande dessinée) lorsque le pointeur de la souris s'arrête assez longtemps sur l'option. Cette zone de texte, appelée info bulle, est utilisée pour éclairer l'utilisateur sur la fonction d'une case de menu, sans qu'il ne la choisisse en cliquant.

Cette fonctionnalité très appréciable a été ajoutée au  $C^3$  en utilisant des étiquettes de texte flottantes dans l'espace 3D. Une étiquette apparaît proche de la case sur laquelle le pointeur s'arrête assez longtemps, c'est à dire au bout d'un délai de une à deux secondes et disparaît dès que le pointeur change à nouveau de case. Elle se présente comme une zone de texte plate, et est construite comme un *billboard*. Un billboard est un terme graphique désignant une texture qui maintient en permanence son alignement d'origine avec la direction de regard de l'utilisateur. Ici notre texture contient le texte descriptif et s'aligne perpendiculairement à la direction de regard, pour rester parfaitement lisible quelque soit la position de la tête de l'utilisateur ou ses mouvements.

## 3.9 Conclusion

Nous avons abordé le contrôle d'application en environnement virtuel sous l'angle original des raccourcis. Ce mécanisme connu sur station de travail sous le nom de raccourcis clavier se distingue des menus traditionnels en s'attachant à contrôler un jeu réduit de commandes, plus rapidement. Il n'existe pas à notre connaissance d'équivalent de cette approche pour les environnements virtuels. Une des difficultés du contrôle d'application en environnement virtuel vient de la pauvreté en nombre de signaux des périphériques

d'interaction. Il n'existe pas d'équivalent du clavier, cependant de nombreux équipements d'interaction peuvent jouer le rôle d'une souris dans un univers 3D. Nous avons donc construit notre solution sur l'extension à un univers 3D d'une technique concurrente des raccourcis claviers, s'appuyant sur l'utilisation de la souris : les *Marking Menus*.

Le *Command and Control Cube* ou  $C^3$  étend le concept des *Marking Menus* à un univers à trois dimensions. Il propose une interface graphique sous forme cubique, divisant l'espace en 3x3x3 cubes adjacents. Combinant une technique de sélection dans un menu graphique à une technique de sélection par geste, il permet au sein du même paradigme de s'adapter au niveau d'expérience des utilisateurs : novice ou expert. Un soin particulier a été porté aux considérations d'obstruction visuelle, particulièrement importante dans les environnements virtuels. Le  $C^3$  répond à ces exigences par son apparition "intelligente" selon la direction de regard de l'utilisateur, sa transparence, sa forme homogène et condensée, et son placement commandé par la position de la main de l'utilisateur.

La manipulation du  $C^3$  est assurée par la main non dominante pour ne pas interrompre le déroulement du travail réel d'un utilisateur dans un univers virtuel, généralement réalisé par la main dominante. Elle est facilitée par des gestes simples : sélection par des mouvements rectilignes de la main dans des directions simples de l'espace, sans considération d'amplitude, et un certain nombre d'indices visuels et proprioceptifs : alignement des axes du  $C^3$  sur les axes haut-bas, gauche-droit, avant-arrière du corps de l'utilisateur, et sur les bords des écrans du Plan de Travail Virtuel.

Une évaluation approfondie du  $C^3$  auprès d'utilisateurs novices a permis de valider l'approche proposée, et de comparer le fonctionnement  $C^3$  sous trois modalités de retour sensoriels différentes et en l'absence de tout retour (mode aveugle). Les retours tactiles et sonores n'ont pas été convaincants pour améliorer les performances du  $C^3$ , mais le mode aveugle a déjà des performances encourageantes. La répartition des performances en vitesse d'exécution et en taux d'erreur, dresse une carte d'accessibilité des différentes cases et donne des indices pour le meilleur placement des options d'une application dans le menu  $C^3$ . Cette carte apporte un éclairage sur les difficultés rencontrées par les sujets. Elle note une difficulté pour atteindre les cases demandant les flexions et extensions du bras les plus importantes, et pour le plan inférieur du  $C^3$ . La présence proche de l'écran inférieur du Plan de Travail Virtuel incitant les sujets à retenir le mouvement de leur bras en est probablement la raison.

Suite à cette évaluation, le  $C^3$  a été étendu à une version hiérarchique pour transformer ce système de contrôle d'application à accès rapide en un menu complet, capable d'accéder à un nombre arbitrairement grand d'options. Dans notre conception du modèle hiérarchique pour le  $C^3$ , nous respectons le paradigme originel d'interaction, préservant les avantages de rapidité et de discréetion visuelle du  $C^3$ . Un affichage permet de suivre la liste des cases choisies, dans l'arbre hiérarchique des options de menu.

Le  $C^3$  propose ainsi une adaptation pour environnement virtuel de deux grands domaines du contrôle d'application sur station de travail : les raccourcis et les menus hiérarchiques. Les menus contextuels forment un dernier volet populaire des menus linéaires de l'interface WIMP. Complétant les deux approches précédentes, nous proposons une version contextuelle du  $C^3$  s'appuyant sur la structure des deux versions précédentes.

Enfin, nous apportons un équivalent 3D des infobulles, des étiquettes flottantes de texte décrivant la fonction de l'option de menu pointée, pour aider à l'exploration du menu  $C^3$ .

Le système de contrôle d'application  $C^3$ , très modulaire, a été porté avec succès sur d'autres plateformes, dont OpenMASK de l'IRISA de Rennes, Open Inventor Immersif de la compagnie TGS. Il a également été réutilisé par le centre de recherche *Advanced Virtual Prototyping* de l'université de Salford en Angleterre.

Le  $C^3$  fonctionne sur le Plan de Travail Virtuel. Il peut être directement exploité avec un mur virtuel sans modification. Il peut également facilement être porté sur d'autres configurations de réalité virtuelle, en examinant la question de son orientation dans la scène. Le  $C^3$  est naturellement aligné sur les bords des écrans du Plan de Travail Virtuel. Pour des configurations plus immersives qui ne possèdent pas de repères physiques, il est nécessaire de considérer le meilleur alignement pour le  $C^3$  : par rapport au bras, aux épaules ou à l'orientation de la tête de l'utilisateur par exemple.



# Chapitre 4

## Saisie de chaînes de caractères

### 4.1 Introduction

Les configurations de réalité virtuelle promettent de grands bénéfices pour la visualisation et la manipulation de maquettes virtuelles. Nous avons vu dans les chapitres précédents que les bouleversements d'un contexte immersif qui apporte à la fois une troisième dimension à gérer et une interaction directe nécessitent de réévaluer pour ces configurations les techniques d'interaction classiques du bureau. Dans ce nouveau contexte, les différences se portent également sur les périphériques d'interaction disponibles et des tâches aussi communes que la saisie de chaînes de caractères doivent être à leur tour reconsidérées.

Les environnements virtuels se prêtant plus à la visualisation et la manipulation de données 3D, il ne s'agit pas ici de concurrencer des systèmes complexes de traitement de texte, plus adaptés aux stations de travail standards, mais de fournir un mécanisme minimal et ergonomique pour la saisie de caractères alphanumériques et spéciaux pour des opérations courantes comme l'enregistrement d'un travail sous un nom de fichier ou la saisie de valeurs numériques précises.

L'état de l'art a montré comment le contexte particulier des environnements virtuels immersifs prohibe l'usage traditionnel du clavier, du fait de contraintes physiques et d'encombrement spatial. Une première solution sera présentée dans ce chapitre mettant en

---

œuvre un clavier entièrement virtuel, semblable à un panneau de contrôle. Les mécanismes d'interaction entre l'utilisateur et un clavier virtuel sont cependant différents des mécanismes d'un clavier réel et le transfert de compétences n'est pas direct.

Les caractéristiques indispensables à un système de saisie de chaînes de caractères pour environnement virtuel s'apparentent sur plusieurs points à celles rencontrées pour les techniques d'écritures sur les équipement assistants personnels digitaux (*Personal Digital Assistant* ou PDA en anglais). Suivant les pistes ouvertes par ce constat, nous nous intéresserons tout particulièrement à une technique populaire développée sur PDA par Ken Perlin [94], nommée QuikWrite. Reprenant le concept d'écriture au stylet de QuikWrite, nous développons une nouvelle technique de saisie pour environnement virtuel utilisant le stylus, baptisée QuikWrite VR [56]. Le chapitre se termine par des évaluations informelles de cette technique.

## 4.2 Le clavier virtuel

Comme nous l'avons vu dans l'état de l'art, la première solution qui consiste à laisser l'utilisateur entrer du texte à l'aide d'un clavier standard n'est pas souhaitable pour les environnements virtuels. Si le périphérique est adapté pour la saisie de texte sur station de travail, les conditions d'utilisation différentes en environnement virtuel l'excluent comme solution viable. Des équipements dédiés et miniaturisés peuvent alors remplacer le clavier traditionnel pour la saisie de chaînes de caractères. Cependant, ces équipements ne permettent pas un transfert de compétences complet depuis le clavier traditionnel car il reste préférable de ne pas multiplier les équipements que l'utilisateur doit porter pour interagir avec l'univers virtuel.

Puisqu'il est difficile d'importer un clavier réel dans une configuration de réalité virtuelle, notre première approche consiste à imaginer un clavier virtuel et l'intégrer comme un widget 3D (ou un ensemble de touches, agissant chacune comme un widget 3D) au sein de l'environnement virtuel. Un clavier virtuel peut être disponible à tout instant, n'a pas de poids, ne nécessite pas d'être tenu par un support physique ou tenu en main, et peut disparaître sur commande pour ne pas obstruer la vue.

### 4.2.1 Technique d'interaction

La première approche envisagée pour interagir avec le clavier virtuel suggère de munir l'utilisateur de gants de données pour reproduire la frappe habituelle sur clavier réel. Cependant elle présente plusieurs difficultés concrètes, liées aux exigences de la correspondance monde réel - univers virtuel.

Pour des cibles à portée de main, la sélection directe d'objets virtuels par un médium réel ne permet pas une grande précision. Les raisons sont multiples. Des problèmes de précision [79] dans la correspondance entre la position réelle du médium et sa position donnée par les capteurs dans la scène virtuelle peuvent intervenir. Ils sont dus aux calibrages des capteurs, qui doivent être vérifiés périodiquement. La perception de la profondeur des objets virtuels est dépendante de paramètres variables selon les utilisateurs comme la position et l'écartement supposé des yeux dans le calcul d'une vue stéréoscopique. De petites différences dans la position perçue des objets d'un utilisateur à l'autre peuvent se révéler critiques dans le cas d'un clavier virtuel, pour la localisation des touches. Enfin, les contraintes graphiques exposent les applications à des lenteurs d'affichage ou *lag* en anglais, perturbantes lors des tâches de sélection [119]. Pour ces raisons, la plupart des techniques de sélection suggèrent l'utilisation d'un intermédiaire graphique entre les mondes réel et virtuel, un élément virtuel servant à désigner, pointer des objets de la scène 3D et réduire la tâche à une interaction entre deux objets, virtuels l'un et l'autre.

Cette imperfection de l'adéquation spatiale du monde virtuel et physique, ne permet pas encore de garantir à l'utilisateur un confort de frappe suffisant avec le doigt d'un gant de données ou la pointe d'un stylo équipé d'un capteur. Nous choisissons finalement ici de réutiliser la technique du rayon virtuel pour désigner les touches lors de l'opération de saisie.

### 4.2.2 Placement

Trois méthodes de placement différentes ont été testées pour notre clavier virtuel. Dans chacune, le rayon sert à pointer sur les touches du clavier, et le bouton du stylus, à valider cette sélection. Les touches changent de couleur dès que le rayon les intersecte pour donner un indice visuel à l'utilisateur. Le texte s'affiche au dessus du clavier.

Dans la première technique, le clavier est fixe dans la scène et le rayon libre de se déplacer et choisir les touches. Dans la seconde, le clavier peut être "tenu en main", en cliquant sur un bouton accroché à une bague (dispositif déjà présent pour le *C<sup>3</sup>*). Le capteur de ce dispositif donne la position de la main, et le clavier devient solidaire de ses déplacements tant que le bouton est maintenu enfoncé. Le rayon se comporte comme auparavant, mais cette fois l'utilisateur emploie les deux mains pour choisir les touches du clavier.

Enfin dans la troisième et dernière méthode, l'orientation du clavier est imposée par la position de l'utilisateur. Le clavier est orienté à chaque instant pour faire face à son regard, déterminé grossièrement par un vecteur allant de la position du capteur de tête au centre du clavier. De plus, le clavier est déplacé dans l'espace par les mouvements du rayon. Lorsque le rayon pointe à l'intérieur du clavier, le clavier reste immobile. Dès lors qu'il atteint un bord du clavier, celui-ci est "entrainé" dans la direction voulue, de manière à ne jamais laisser le rayon pointer hors de son socle. Le clavier évolue ainsi sur une sphère autour de la tête de l'utilisateur. Le rayon de cette sphère, c'est à dire la distance du clavier à la tête de l'utilisateur, est réglé par la position de la main tenant le stylo. La distance stylus-clavier reste constante. Ainsi, lorsque l'utilisateur étend son bras, le clavier s'éloigne. Lorsqu'il le ramène contre lui, le clavier se rapproche. Dans cette dernière méthode, le clavier ne peut jamais être perdu hors du champ de vision de l'utilisateur ; il est toujours présent au bout de son bras tenant le stylo, et orienté pour faire face à son regard.

#### 4.2.3 Evaluations informelles

Chacune des trois méthodes de placement ont été testées auprès d'utilisateurs novices, ayant cependant une expérience préalable de la configuration de réalité virtuelle employée, en l'occurrence le Plan de Travail Virtuel. La méthode générale est jugée fonctionnelle mais lente, du fait de l'écriture par sélection des touches au rayon, qui s'apparente à une écriture à un seul doigt. La précision requise pour sélectionner au rayon est un deuxième facteur ralentissant la saisie. Les performances des deux premières méthodes sont perçues comme très proches, en dépit de la possibilité pour la seconde de faire collaborer les deux mains. La troisième méthode prend en charge les problèmes de placement mais reste identique au niveau de la rapidité et de la précision de sélection.

#### 4.2.4 Conclusion

La technique du clavier virtuel permet de simuler un clavier réel dans un environnement virtuel, sans subir les contraintes d'encombrement et de disponibilité. Une difficulté principale de la saisie de texte, liée à la faiblesse du nombre de signaux discrets des équipements de réalité virtuelle, est contournée en intégrant l'avatar d'un clavier réel dans la scène virtuelle. L'interaction suit la méthode du rayon virtuel pour éviter les problèmes de correspondance entre les mondes réel et virtuel, qui prohibe une frappe directe avec les doigts. En contrepartie, la méthode de saisie est sensiblement plus lente que la saisie traditionnelle au clavier.

### 4.3 QuikWrite VR

#### 4.3.1 Introduction

Nous venons de voir une première approche cherchant à intégrer le paradigme du clavier dans l'univers virtuel. Cette approche suggère une répartition spatiale des caractères, avec un bouton ou presque par symbole à sélectionner. Cependant les équipements de réalité virtuelle sont souvent pauvres en nombre de signaux discrets ou boutons, en comparaison du nombre de touches d'un clavier réel. Les périphériques pour l'interaction sont communément composés d'un capteur et d'une poignée de boutons. Le stylus est un équipement populaire pour ces configurations, qui bien que ne présentant pas de nombreux signaux d'entrée comme un clavier, possède une forme de stylo qui évoque l'écriture. Notre seconde approche consiste ici à développer une technique d'interaction fondée sur cette idée.

Nous remarquons dans un premier temps une similitude avec un autre domaine de l'écriture sur équipement mobile. En effet, un mode d'écriture digitale à l'aide d'un stylo existe également sur un autre type de support : les assistants numériques personnels (PDA en anglais). Le contexte d'utilisation présente des similitudes intéressantes. La saisie de texte, par le biais d'un stylo de petite taille appelé stylet, partage les mêmes exigences de faible encombrement total et de mécanisme simple et rapide pour la prise de notes. Dans les deux cas, l'objectif recherché est moins de fournir un système élaboré de traitement de



FIG. 4.1 – Un PDA (Palm V<sup>TM</sup>), copyright Palm inc.

texte complet et élaboré qu'un médium pour l'enregistrement simple et rapide de courts textes constitués de caractères alphanumériques ou spéciaux.

L'écriture au stylet est souvent limitée à une zone sensible spécifique de l'écran. Plusieurs techniques de saisie existent, interprétant différemment les signes dessinés sur la surface sensible. L'écriture graffiti<sup>TM</sup>, très proche du dessin de lettres majuscules, est une technique courante pour les Palms<sup>tinyTM</sup>. Une technique d'écriture particulière conçue par Ken Perlin en 1998, nommée QuikWrite [94], a suscité notre intérêt pour sa simplicité de concept et de réalisation, et sa similitude graphique avec le *Command and Control Cube*. Nous proposons de l'adapter aux environnements virtuels. Nous présentons tout d'abord le paradigme d'écriture QuikWrite de Ken Perlin, puis dans une seconde partie notre adaptation de cette technique à un environnement virtuel, ainsi que les extensions que nous avons apporté à la technique, et de premières évaluations informelles.

### 4.3.2 QuikWrite

QuikWrite est un paradigme d'écriture pour la saisie rapide de notes sur Palm Pilot<sup>TM</sup> (cf. figure 4.1). L'utilisateur dispose d'une petite zone d'écriture plate, carrée et d'un stylet. La saisie de texte est réalisée en dessinant sur la surface sans jamais relever le stylet. Les mouvements de la pointe du stylet sur la surface sont interprétés comme si la



FIG. 4.2 – Les 9 zones de QuikWrite

FIG. 4.3 – Un PDA (Palm V<sup>TM</sup>) avec un masque pour QuikWrite, copyright Palm inc.

surface carrée était subdivisée selon une grille de trois par trois (voir figure 4.2).

Ces 9 zones ne sont pas visualisées, car en général les PDA ne permettent pas d'affichage dans la surface de saisie. Il est possible cependant de superposer un masque sur cette petite zone pour palier à ce problème (voir figure 4.3).

Chaque caractère est saisi en commençant le mouvement avec le stylet pointant sur la zone centrale de la grille. Le stylet est déplacé vers les zones adjacentes sans jamais quitter la surface, avant d'être ramené dans la zone centrale. La lettre écrite est définie par les zones visitées. En combinant la première zone atteinte en sortant de la case centrale et la dernière zone visitée avant de revenir en case centrale, on obtient un tableau à deux dimensions de combinaisons possibles. Chaque élément de ce tableau est un caractère (lettre, symbole ou chiffre). L'écriture est réalisée en traçant des séquences de boucles

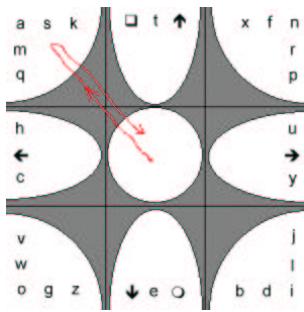


FIG. 4.4 – Trajet du stylo sur la surface de sélection de QuikWrite pour la lettre A.

traversant la case centrale. Par exemple, si le stylet part de la zone centrale, monte dans la zone en haut au milieu puis revient dans la zone centrale, la lettre "t" est sélectionnée. Cette écriture est moins fatiguante que l'écriture Graffiti [10] standard du Palm, car elle s'apparente à l'écriture manuscrite par sa caractéristique de ne pas avoir à relever le stylet pendant la saisie. En contrepartie elle demande un temps d'apprentissage, estimé à deux à trois heures par l'auteur, pour mémoriser l'association des lettres et des mouvements sur cette grille fictive.

### 4.3.3 Principe d'écriture

L'écriture est réalisée par une série de boucles débutant et finissant en zone centrale. La boucle la plus simple consiste à quitter la zone centrale pour une zone adjacente et y revenir tout de suite. Les 8 zones entourant la zone centrale donnent ainsi accès à 8 caractères. A titre d'exemple, la lettre "a" est obtenue en déplaçant le stylet de la zone centrale vers la zone en haut à gauche du panneau, et en revenant en zone centrale (figure 4.4). Sur le panneau, la lettre "a" figure dans le coin de la zone, dans l'axe de la diagonale.

Pour étendre ce jeu de 8 caractères, il est possible de traverser deux zones adjacentes avant de revenir en zone centrale. Cette fois, par exemple, pour sélectionner la lettre "s", l'utilisateur traverse en premier la zone de la lettre "a" puis celle qui est immédiatement à sa droite, avant de revenir en zone centrale (voir figure 4.5). La lettre "s" est inscrite dans la même zone que la lettre "a", car il s'agit de la première zone à traverser pour sélectionner un "s". Elle est dans cette zone, immédiatement à droite de la zone du "a", car il faut se rendre dans la zone immédiatement à droite de la zone du "a", avant de revenir en position centrale.

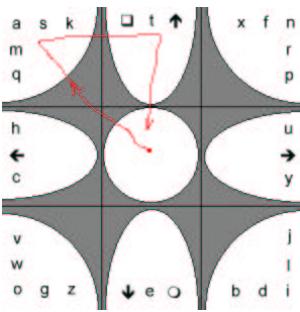


FIG. 4.5 – Trajet du stylo sur la surface de sélection de QuikWrite pour la lettres S.

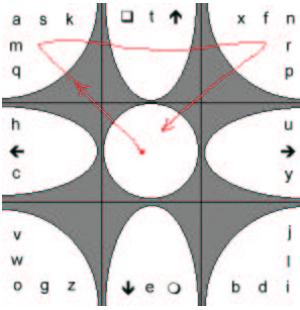


FIG. 4.6 – Trajet du stylo sur la surface de sélection de QuikWrite pour la lettres K.

Enfin, un déplacement d'une longueur de deux zones avant de rejoindre la position centrale est également possible. Pour atteindre la lettre "k", l'utilisateur doit comme précédemment initier son mouvement vers la zone de la lettre "a", puis le poursuivre horizontalement vers la droite sur deux zones de distance, avant de rejoindre le centre (voir figure 4.6). La lettre "k" figure sur le panneau dans la même zone que la lettre "a" car le mouvement débute dans cette zone là. Elle est inscrite à la droite de la lettre "s", elle-même à droite de la lettre "a", pour rappeler que le déplacement du stylo se fait sur deux zones vers la droite.

La disposition des lettres sur le panneau de QuikWrite choisie par Ken Perlin tend à optimiser le déplacement moyen du stylet. Les lettres les plus courantes de l'alphabet anglosaxon sont placées par défaut dans les boucles de sélection les plus courtes (en distance parcourue par le stylo) : les aller retour zone centrale → zone extérieure → zone centrale. La plupart des voyelles se retrouvent ainsi dans les zones diagonales ou dans la zone basse. Par ordre de fréquence dans la langue anglaise les lettres suivantes de l'alphabet sont disposées dans les boucles prenant deux zones de longueur, et en dernier celles prenant trois zones.

Les boucles simples vers la gauche et la droite sont réservées pour deux actions spon-

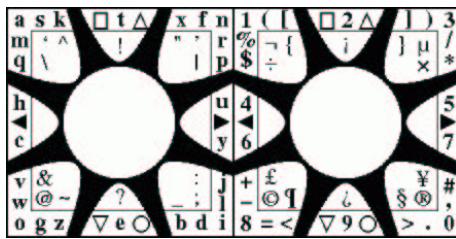


FIG. 4.7 – Les 9 zones de QuikWrite donnent accès à 4 modes : minuscules, majuscules, numérique et ponctuation

tanément attendues par les utilisateurs à cet emplacement : insertion d'un espace (et déplacement du curseur d'écriture vers la droite) et effacement du précédent caractère (et déplacement du curseur d'écriture vers la gauche).

Cette première répartition des lettres sur le cadran de QuikWrite permet d'écrire les 26 lettres minuscules de l'alphabet, plus les touches d'insertion d'espace vers la droite ou la gauche du curseur (*space* et *backspace* en anglais) et de conserver 4 emplacements pour d'autres symboles. Les claviers traditionnels possédant plus d'une centaine de touches différentes directement accessibles et plus encore en tenant compte des combinaisons possibles avec les touches Ctrl, Alt et Shift, il est nécessaire d'étendre le nombre de symboles accessibles par QuikWrite. Pour ne pas se limiter à ce premier jeu simple de caractères, QuikWrite propose un méthode d'accès à trois cadrants supplémentaires : majuscules, caractères numériques et symboles de ponctuation.

Les quatres emplacements disponibles sur le premier cadran sont utilisés pour entrer dans des modes "majuscule", "numérique" ou encore "ponctuation" (cf. figure 4.7). En entrant dans le mode "majuscule", toutes les boucles précédentes permettant d'écrire des caractères minuscules donnent à présent accès à des caractères majuscules. Sur les PDA la zone sensible d'écriture n'est pas une zone d'affichage. Par conséquent seule l'interprétation des mouvements du stylet change, laissant l'utilisateur deviner seul qu'il a changé de mode. Dans chaque cadran supplémentaire, certaines boucles restent réservées à des actions particulières. Les boucles d'aller retour vers la gauche et la droite sont toujours associées à l'espace et l'effacement arrière.

L'utilisation des caractères alphabétiques minuscules constitue dans la plupart des mécanismes de saisie de texte le cœur de l'écriture. L'accès aux caractères spéciaux des panneaux supplémentaires reste ponctuel, et souvent le temps d'un seul caractère comme

pour placer une majuscule en première lettre d'un mot, ou un point à la fin d'une phrase. Au vu de cette remarque, les modes spéciaux, une fois appelés, ne durent que le temps d'une seule saisie. Dès le premier caractère tapé, QuikWrite revient automatiquement au panneau des minuscules. Pour une plus grande liberté, il est possible de verrouiller QuikWrite dans un des modes spéciaux, pour ne pas les quitter automatiquement après une saisie (par exemple, lorsque l'on veut écrire une série de chiffres). En exécutant deux fois la boucle de sélection du mode spécial voulu, une première fois pour le choisir, une seconde fois pour entrer dans le mode verrouillé, l'utilisateur fige le panneau de caractères en cours, qui ne change plus tant qu'une troisième boucle identique n'est pas exécutée pour revenir au panneau des minuscules.

#### 4.3.4 Quikwrite VR

QuikWrite VR est notre adaptation du concept de QuikWrite à un univers virtuel 3D immersif. Le stylet utilisé pour la version PDA est remplacé par le stylus.

Les zones d'écriture de QuikWrite, décrites par Perlin comme une grille trois par trois, rappellent fortement la structure du *Command and Control Cube*. Nous pensons qu'il est important pour l'accessibilité d'une interface, de présenter des paradigmes d'interaction similaires, ainsi que des éléments graphiques d'apparence commune, lorsque cela est possible. Nous avons ainsi l'idée de représenter la surface sensible de saisie comme un objet virtuel de la scène : 3x3 cubes adjacents, flottant dans l'espace selon un plan vertical faisant face à l'utilisateur (voir figure 4.8).

##### 4.3.4.1 Ecriture au stylus

Dans l'hypothèse où le stylus interagit directement avec l'univers virtuel, les difficultés habituelles de correspondance entre la position réelle du stylus et sa position dans la scène virtuelle resurgissent. De petites différences de position induisent des erreurs de manipulation, et dans ce scénario d'utilisation du stylus, la précision de la pointe glissant sur les cases de la surface de QuikWrite VR est fondamentale. Afin de se soustraire à ces contraintes de précision un avatar du stylo est affiché dans la scène pendant la manipulation. Il est activé en pressant une fois le bouton du stylus physique. L'avatar, que nous

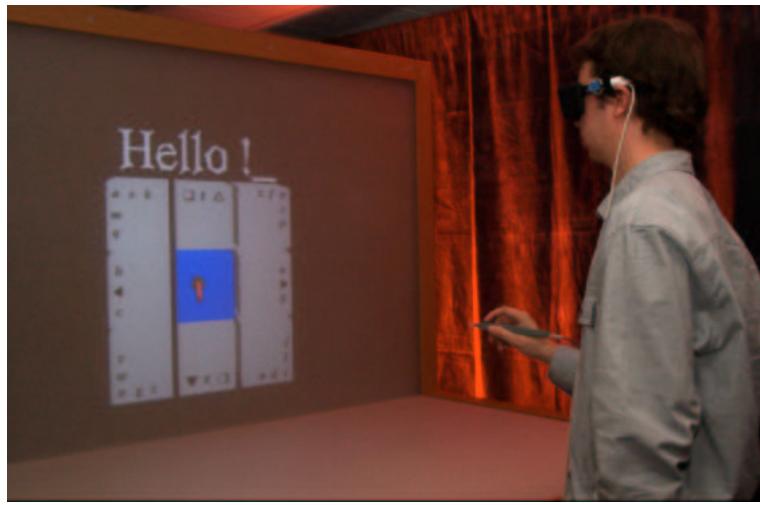


FIG. 4.8 – QuikWrite VR, le système de saisie de chaînes de caractères en environnement virtuel

appelerons stylo virtuel, apparaît perpendiculairement à la surface d'écriture, la pointe touchant la case centrale. Pour le faire disparaître à n'importe quel moment et interrompre la saisie, il suffit de réappuyer sur le même bouton du stylus. Ce contrôle du stylo virtuel permet à l'utilisateur de reposer son bras à n'importe quel moment, en retirant le stylo virtuel de la surface et en désactivant ainsi le mode de saisie.

Les mouvements du stylus guidé par la main de l'utilisateur, sont appliqués au stylo virtuel avec une correspondance un pour un. Cependant, une contrainte est conservée pour s'affranchir des considérations de profondeur et forcer le stylo virtuel à coller à la surface de QuikWrite VR. En filtrant les mouvements du stylus en profondeur, le stylo virtuel est contraint de glisser le long de la surface verticale, qu'il ne peut ainsi quitter. L'utilisateur est plus libre de ses mouvements, car il s'affranchit d'une dimension inutile. Le stylo virtuel fournit un retour visuel à l'utilisateur qui n'est pas directement en train d'écrire sur la surface.

#### 4.3.4.2 Résolution des mouvements diagonaux

Les coordonnées spatiales du stylus sont retranscrites en terme de position dans la grille de cases. Le parcours du stylo virtuel sur la grille est caractérisé par la liste des cases traversées au cours du temps. Nous remarquons lors de premiers essais d'écriture, qu'il est parfois délicat de passer de la case centrale à une des cases occupant un coin, et

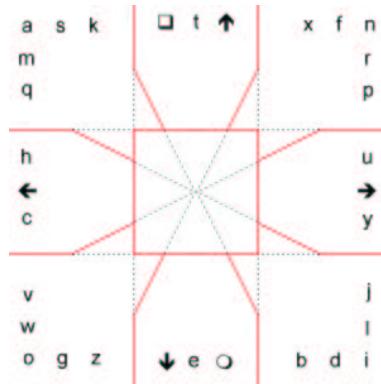


FIG. 4.9 – Division réelle de la surface de travail de notre système QuikWrite

inversement. En effet, la case centrale n'est reliée que par un sommet aux cases diagonales et le geste requis pour ne pas déborder et traverser une case voisine demande une précision excessive. Cette disparité entre les mouvements diagonaux et la traversée de cases, partageant une arête entière comme frontière, n'est pas souhaitable.

Nous remarquons que cette difficulté n'existe pas sur le  $C^3$ . Elle survient ici car QuikWrite VR considère le trajet effectué, et non un instant de sélection.

Afin de faciliter les mouvements diagonaux, nous avons choisi de redéfinir la géométrie des cases. Leur apparence visuelle ne change pas pour ne pas perturber l'utilisateur, seul le traitement déterminant la case effectivement pointée par le stylo à chaque instant est altéré. La nouvelle interprétation des contours de cases est représentée figure 4.9. Ce changement permet de rééquilibrer les angles d'accès aux diverses cases voisines et d'éliminer les précédentes difficultés de sélection.

#### 4.3.4.3 Lisibilité et flexibilité de l'affichage

L'écriture suit la procédure décrite pour QuikWrite, lorsque le stylo virtuel est déplacé sur la surface d'écriture. Sur les agendas électroniques cependant, un utilisateur peut rencontrer deux difficultés à l'écriture avec QuikWrite. QuikWrite VR propose une solution pour chacune d'elle.

Tout d'abord, la zone d'écriture sur un PDA, déjà de petite taille, est subdivisée encore en 9 zones distinctes pour le fonctionnement de Quikwrite. Quikwrite VR dispose d'un espace beaucoup plus large pour dimensionner la zone d'écriture, dont la taille virtuelle

est entièrement paramétrable. En agrandissant les cases, QuikWrite VR réduit considérablement les exigences de précision des mouvements et augmente le confort de l'utilisateur.

D'autre part, Quikwrite demande à l'utilisateur de connaître ou d'apprendre la disposition des lettres sur la grille pour pouvoir écrire. La zone sensible des PDA ne permet généralement pas l'affichage de la grille de cases. Lorsqu'un masque est superposé au PDA, de trois à cinq lettres sont affichées dans un neuvième d'une zone d'écriture étroite. La lisibilité des lettres reste par conséquent limitée et l'efficacité de ce rappel mnémotechnique en est affecté. L'apparition de QuikWrite VR sous forme d'un objet 3D dans l'environnement virtuel, permet d'afficher en texture large et lisible sur les cases ces informations cruciales. L'utilisateur voit la grille sous le stylo virtuel, ce qui lui permet de doser l'amplitude de ses mouvements, et il voit également les lettres sur le bord de la surface d'écriture, disposées comme sur un cadran.

La disposition des lettres sur le panneau de QuikWrite VR est celle choisie par Ken Perlin pour QuikWrite. Comme pour QuikWrite, QuikWrite VR propose l'accès à trois cadraux supplémentaires : majuscules, caractères numériques et symboles de ponctuation. Les quatre emplacements réservés sur le premier cadran sont utilisés pour entrer dans les modes "majuscule", "numérique" ou encore "ponctuation" et utilisent les mêmes symboles : triangle, carré, cercle. En appelant le symbole adapté sur le panneau de QuikWrite, celui-ci est changé pour correspondre au nouveau mode. Toutes les textures appliquées sur les cases sont remplacées pour correspondre au mode actif. En entrant dans le mode "majuscule", toutes les boucles précédentes permettant d'écrire des caractères minuscules donnent à présent accès à des caractères majuscules. De plus toutes les lettres minuscules affichées sur le bord des cases sont également remplacées par des lettres majuscules. QuikWrite VR apporte ainsi un retour visuel à l'utilisateur sur le mode courant activé.

Les trois mode spéciaux, "majuscule", "numérique" et "ponctuation" sont appelés par le mécanisme de Quikwrite en deux étapes. Un mode spécial ne dure que le temps d'une saisie. S'il est appelé deux fois de suite, QuikWrite se verrouille dans ce mode. Un troisième appel redébloque QuikWrite et le ramène dans le mode "minuscule". Pour distinguer ces deux états dans QuikWrite VR, nous changeons l'apparence du symbole désignant l'accès aux panneaux de caractères spéciaux. Chacun des trois panneaux est représenté par une forme géométrique simple : un triangle pour les majuscules, un carré pour les symboles numériques, et un cercle pour les symboles de ponctuation. A la première sélection, la

forme géométrique est à moitié remplie de noir. A la seconde sélection (verrouillage) la forme est entièrement remplie de noir. A la troisième sélection (retour au panneau des minuscules), la forme est vidée à nouveau. L'utilisateur sait ainsi à chaque instant dans quel mode il se trouve.

#### 4.3.4.4 Evaluation informelle

Nous avons conduit une évaluation informelle avec 5 utilisateurs novices. Tous sont familiers des ordinateurs et de la configuration de réalité virtuelle utilisée, ici le Plan de Travail Virtuel. Ils reçoivent en premier lieu une explication du fonctionnement de QuikWrite VR. Puis nous les laissons pratiquer quelques manipulations basiques et écrire quelques phrases pour les entraîner à ce nouveau paradigme d'écriture. Les premières phrases sont tapées lentement. La cause principale est le temps d'apprentissage nécessaire pour la disposition des lettres sur le cadran. Une fois cette première étape franchie, des accélérations sur certains enchaînements de lettres, c'est à dire de boucles, commencent à apparaître et l'écriture devient plus fluide et rapide.

Un des utilisateurs note que l'orientation verticale de la surface est fatiguante pour la main et le bras. Une meilleure orientation pour la surface consiste à l'incliner horizontalement ou lui donner simplement une légère pente. Un autre utilisateur a regretté l'absence de technique rapide de déplacement dans le texte, sans effacer ce qui est déjà écrit. Une utilisation du stylo prolongé d'un rayon virtuel pour repositionner un curseur d'écriture dans le texte serait une solution envisageable. Enfin, une des gênes exprimée a concerné l'impossibilité d'annuler son mouvement, lorsqu'une boucle est initiée de façon erronée mais n'est pas encore finie, i.e. l'utilisateur a déplacé le stylo virtuel hors de la case centrale sans toutefois revenir dans celle-ci. L'utilisateur a suggéré de permettre au stylo virtuel de quitter la surface dans un tel cas de figure. Actuellement, il est contraint de terminer son mouvement et corriger son erreur en effaçant le dernier caractère affiché.

Tous les utilisateurs ont déclaré avoir apprécié cette méthode d'écriture à la fin de leur période d'essai.

## 4.4 Conclusion et perspectives

Les difficultés de la saisie de chaînes de caractères en environnement virtuel viennent des contraintes d'encombrement des équipements accessibles, et de la faiblesse générale en nombre de signaux discrets des périphériques d'interaction. Nous avons présenté une première approche visant à intégrer un clavier virtuel dans l'environnement, en tant qu'objet 3D de la scène virtuelle, afin de s'affranchir des contraintes d'encombrement. Les limitations techniques sur les possibilités d'interaction directe entre le monde réel et le monde virtuel nous ont poussés cependant à envisager une interaction suivant la technique du rayon virtuel. La technique du clavier virtuel, bien que fonctionnelle est cependant assez lente.

Une seconde solution a été envisagée, après constatation de ressemblances entre les contraintes d'une solution de saisie de chaînes de caractères en environnement virtuel et les équipements électroniques mobiles de type PDA. En nous basant sur la technique d'écriture pour PDA nommée Quikwrite, nous avons développé une méthode de saisie similaire pour environnements virtuels : QuikWrite VR. L'interaction s'appuie sur la manipulation du stylus. Un avatar de stylo, contraint à rester sur la surface d'écriture et contrôlé par les mouvements du stylus, affranchit l'utilisateur des problèmes de précision. Son apparition et sa disparition de la scène sont ordonnées par le bouton du stylus, et permettent d'interrompre ou de reprendre la saisie à tout moment. QuikWrite VR tire profit d'une représentation graphique de la grille d'écriture, absente dans le QuikWrite originel, pour afficher des informations visuelles sur la disposition des lettres et le choix des différents ensembles de caractères disponibles. Sa largeur permet une meilleure aisance d'écriture. Le système de saisie de texte Quikwrite VR a été développé sur la plate-forme Minifly de l'équipe i3D de l'INRIA et porté avec succès sur Open Inventor Immersif dans le cadre du projet RNTL "Open Inventor Immersif" [27].

Des évaluations informelles ont suivi ces deux approches. Nous prévoyons dans l'avenir de mener une évaluation plus complète et précise de l'outil d'écriture QuikWrite VR, spécialement en comparaison des autres méthodes d'écriture basées sur un stylet l'ayant inspiré. L'apport d'une surface physique solide pour l'affichage de la grille de QuikWrite VR est également considéré comme source de confort et aide à l'écriture. La palette virtuelle de Coquillart et al. [31] peut être à cet égard une solution intéressante pour des configurations exploitant déjà ce périphérique pour d'autres tâches d'interaction.

# Chapitre 5

## Exploration de scènes 3D à l'aide de la palette virtuelle

### 5.1 Introduction

Les configurations de réalité virtuelle créent des environnements 3D immersifs qu'un ou plusieurs utilisateurs peuvent parcourir et explorer. Comme dans le monde réel un utilisateur peut choisir d'inspecter son environnement en se déplaçant dans la scène, autour des objets virtuels. Les techniques de navigation, présentées dans l'état de l'art, permettent de changer le point de vue sur les objets et les paysages. La modification du point de vue n'est cependant pas la seule méthode envisageable pour explorer et découvrir son environnement virtuel. Dans le cas de scènes de grande taille ou complexes, des méthodes complémentaires pour l'exploration de données spatiales peuvent grandement accroître les performances concernant la découverte et l'analyse de données 3D. Elles cherchent par exemple à traiter efficacement les objets distants, les objets cachés derrière des obstacles opaques ou l'intérieur d'objets creux. Elles peuvent également proposer des vues supplémentaires ou améliorées sur les objets de la scène, pour les observer sous des angles inhabituels, pour changer leur apparence ou leurs paramètres de rendu graphique. Dans la première partie de ce chapitre nous présentons un état de l'art des différentes techniques augmentant les capacités de vision d'un utilisateur immergé. Ces techniques étendent ou améliorent les sens de l'utilisateur dans un environnement virtuel. Ces fonc-

---

tionnalités doivent également s'intégrer naturellement dans l'environnement pour ne pas déconcerter l'utilisateur. Des paradigmes d'interaction sont attendus pour réaliser cette intégration et procurer les méthodes de contrôle nécessaires.

Nous présentons dans ce chapitre une solution pour l'exploration de scènes 3D s'appuyant sur l'utilisation de la Palette Virtuelle, un outil d'interaction composé d'un manche tenu en main et d'une surface plate transparente [31], répérée par un capteur de position. Après une présentation des techniques existantes autour de ce périphérique, nous exposons une nouvelle métaphore s'appuyant sur la palette : la métaphore du miroir.

L'outil ainsi créé est baptisé *Magic Mirror* [53]. Les capacités d'exploration du *Magic Mirror* sont ensuite étendues par l'ajout de nombreuses fonctionnalités développant la plupart de ses caractéristiques : champ de vision, réflexion, luminosité, élimination des obstacles opaques, etc. Nous présentons enfin un ensemble de tâches d'exploration que nous avons identifiées, pour lesquelles le *Magic Mirror* est particulièrement adapté.

Dans un deuxième temps, nous verrons comment ces fonctionnalités et celles des techniques classiques des filtres 2D peuvent être intégrées, sans effet miroir cette fois et de manière directe sur la palette virtuelle, celle-ci permettant de localiser les filtres dans l'espace sur un outil aisément manipulable par l'utilisateur. Cette seconde approche sera baptisée *Magic Windows*. Nous terminerons par les premiers résultats des évaluations informelles de ces techniques.

## 5.2 Exploration de scènes 3D et utilisation des filtres

Il existe de multiples techniques pour améliorer ou faciliter l'exploration de données graphiques. Nous nous intéressons ici à quelques unes de ces techniques, principalement celles utilisant le concept de filtres, modifiant localement ou globalement la vue des éléments d'une scène, que nous reprendrons dans les sections suivantes.

En accédant aux données spécifiques aux applications, et en proposant des modes de visualisation intégrés, les filtres offrent un regard différent sur tout ou partie d'une scène. Utiliser un filtre pour changer la manière dont les données sont visualisées est un concept ancien, déjà introduit par le passé pour des applications 2D [123, 50, 70]. Cette technique d'interaction ouvre la voie à toute une gamme d'outils d'exploration. Evolutive, elle s'enrichit de l'ajout de chaque nouveau filtre. Le nombre de filtres développables

n'est limité que par les besoins de l'application. Diverses approches ont été suivies pour l'intégration naturelle de ces filtres au sein de l'application. Nous distinguons deux étapes importantes à l'utilisation des filtres en environnement virtuel :

- L'intégration du paradigme à un univers 3D et les adaptations nécessaires du concept à l'ajout de la troisième dimension.
- Les méthodes de contrôle associées pour manipuler le "filtre 3D" dans un univers immersif.

### 5.2.1 Intégration des filtres en 3D

Bier et al. proposent en 1993 un système de filtre ingénieux nommé *Magic Lenses*. Ces lentilles magiques sont des widgets 2D associant un comportement à un élément graphique. Elles se présentent sous la forme de zones rectangulaires transparentes similaires à des carreaux de vitres, recouvrant les données et permettant soit de changer la façon de visualiser ce qu'elles recouvrent, soit de changer la fonction d'un clic souris sur ce qu'elles recouvrent. Les carreaux transparents associés à des fonctionnalités passives transforment le contenu visible à travers cette petite fenêtre rectangulaire, comme par exemple une lentille grossissante. Les autres, associés à des fonctionnalités actives changent cette fois l'interaction à l'intérieur de leur fenêtre. Les événements clic souris sont interprétés par la lentille. Par exemple, un clic peut peindre en bleu une forme géométrique pointée à l'intérieur d'une lentille ayant cette fonction. Le système est modulaire et permet de superposer plusieurs lentilles de fonctions différentes pour obtenir des macro-opérations. Bier et al. décrivent des exemples d'utilisation parmi la multitude imaginable : travail sur les formes, les propriétés, la couleur, zone tampon pour des opérations copier/coller, prévisualisation de changements, etc. Les variations possibles dans la conception d'outils transparents sont explorées plus largement dans un second article où les auteurs décrivent une taxonomie [7] pour cette gamme d'outils.

Cignoni et al. développent l'année suivante un outil d'exploration 3D, s'appuyant sur le concept des *Magic Lenses*. Les *Magic Spheres* [28] sont construites sur la métaphore d'une sphère transparente, traversant les objets comme un fantôme, et modifiant l'apparence de leur espace intérieur, i.e. tout ce qui se trouve inclus dans le volume de la sphère. La manipulation reste simple pour l'utilisateur, qui n'a à contrôler que le rayon de la sphère

et la position de son centre. Pour renforcer la sémantique d'une lentille de verre, Cignoni et al. ont également recréé un effet de bord flou, pour les éléments présents à la périphérie du globe. A titre d'exemple, ils présentent pour les *Magic Spheres* toute une gamme de fonctions de filtrage : changement du niveau de détail des éléments graphiques, affichage exclusif du maillage, loupe, modification locale des paramètres de rendu, interpolation de surfaces à partir du maillage initial, etc. Ils remarquent également que la composition de filtres 3D par la juxtaposition des sphères peut être problématique au niveau de l'ordre d'application et des opérations de placement de ces différents filtres pour cerner la région désirée. A ce titre, ils préconisent l'association de plusieurs filtres à une unique sphère, pour préserver la simplicité d'interaction.

Viega et al. [118] reprennent également les *Magic Lenses* pour les étendre à l'espace à trois dimensions. Ils proposent deux approches différentes, selon que les *Magic Lenses* sont intégrées dans l'espace comme objet 2D à part entière, ou que leur forme est retravaillée pour s'adapter à un environnement possédant une dimension supplémentaire. Dans le premier cas, les *Flat Lenses* ou lentilles plates sont des lentilles 2D à travers lesquelles l'utilisateur voit le monde différemment. La zone concernée par cette modification n'est plus un rectangle plat mais une pyramide ayant la lentille comme petite base, et appelée *lens frustum*. Dans le second cas, Viega et al. imaginent des lentilles volumétriques ayant une forme géométrique à six faces. Le volume compris à l'intérieur de ces 6 plans définit la zone de l'espace sur laquelle s'appliquent les fonctions de filtrage. Ils notent également pour ces lentilles volumétriques l'absence d'une sémantique forte comme en 2D pour la notion de recouvrement, nécessitant des choix arbitraires sur l'ordre d'application des filtres se recouvrant.

### 5.2.2 Techniques de contrôle des filtres 3D

Les solutions précédentes se concentrent en premier lieu sur l'intégration du concept du filtre dans un univers 3D, souvent non immersif, et les méthodes de contrôle de ce filtre ne constituent pas le point focal de ces études.

Wloka illustre une utilisation de son périphérique *Virtual Tricorder* avec une lentille magique plate grossissante (voir figure 5.1). La lentille est ancrée dans l'espace à la position du tricorder. Cette technique de contrôle de la lentille permet de faire varier interactive-

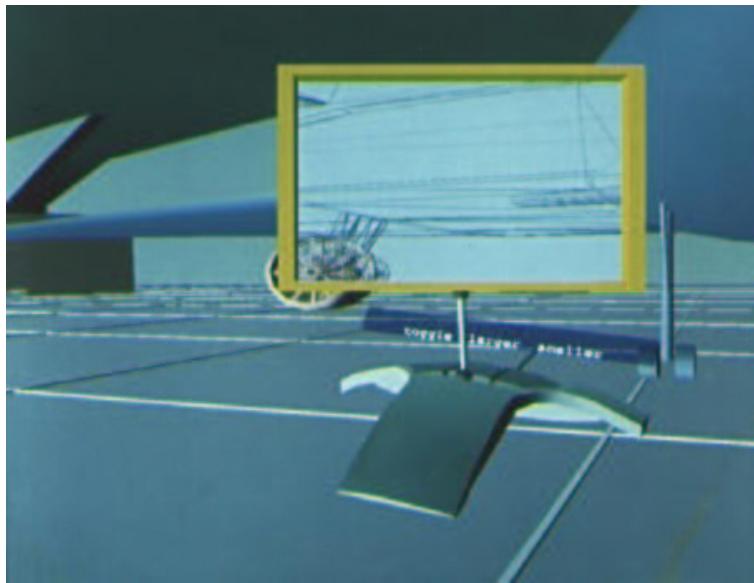


FIG. 5.1 – Utilisation d'une fenêtre filtrante sur les données de la scène.

ment la largeur du champ de vision modifié en rapprochant ou éloignant le tricorder de ses yeux.

Mine propose une technique analogue pour une lentille grossissante [88], n'employant aucun outil physique. Les mains de l'utilisateur définissent par leur position spatiale les bords opposés d'un rectangle, à la manière d'un réalisateur de film cadrant une scène. La taille du rectangle définit le facteur de zoom. L'utilisateur décide de basculer de la vue normale à la vue grossissante en traversant le plan du rectangle, et inversement. Cette technique est baptisée *Head-Butt Zoom*.

Enfin, Stoev et al. [114] présentent en 2002 une taxonomie des différentes approches de la métaphore *Through-The-Lens* (à travers la lentille) pour laquelle une simple seconde vue sur le monde virtuel est proposée.

## 5.3 Solutions proposées

### 5.3.1 Introduction

Les techniques d'exploration proposent des fonctionnalités améliorant ou étendant la vision humaine. Ces fonctionnalités peuvent se manifester et être manipulées de façons très variées pour servir les buts de l'utilisateur.

- Elles peuvent être communes à tout l'univers virtuel et fonctionner selon le principe de modes. Les modes sont combinables entre eux mais ne sont pas spécifiques à un lieu et affectent sans distinction tout objet de l'espace, agissant comme un contexte de travail.
- Elles peuvent être attachées aux différents objets de la scène, reliant sémantiquement chaque objet aux fonctions disponibles pour mieux le visualiser.
- Elles peuvent être associées à un objet spécifique de l'espace, manipulable et qui permet d'agir sur tout autre objet sur lequel il est utilisé.

Cette dernière approche correspond à la métaphore<sup>1</sup> de l'outil. L'outil concentre en un objet de l'espace une apparence et une fonctionnalité. Les techniques d'interaction fondées sur la métaphore de l'outil sont particulièrement attractives dans les environnements virtuels immersifs, car les outils sont déjà un médium d'interaction très courants dans le monde réel. Leur utilisation est naturelle pour l'homme, et peut être très intuitive si l'outil utilisé pour l'interaction est une métaphore d'un outil réel, car elle fournit alors immédiatement à l'utilisateur un modèle cognitif de l'outil. Il s'agit ici d'une seconde métaphore s'imbriquant dans notre métaphore initiale de l'outil. L'outil virtuel ainsi décrit s'appuie alors simultanément sur la métaphore de l'outil, conférant des fonctionnalités à un objet de l'espace manipulé à la main, et sur la métaphore d'un outil réel, lui apportant un modèle cognitif pour sa fonction et sa manipulation.

Les techniques de filtres 2D exportées dans un univers 3D forment de puissants outils d'exploration de scènes complexes. Dans un environnement virtuel immersif, il est nécessaire de développer à la fois des méthodes d'intégration de la métaphore des filtres à un univers 3D mais également de fournir conjointement des techniques de manipulation pour

---

<sup>1</sup>voir les travaux de Brooks pour une discussion sur les métaphores [22]



FIG. 5.2 – Le *Virtual Remote Control* sur la palette virtuelle de Coquillart et Wesche [31]

contrôler ces "filtres 3D". Nous présentons dans cette partie notre solution pour l'exploration de scènes 3D, attachant des filtres 2D à la surface plate d'une palette virtuelle [31] pour recréer des outils du monde réel (loupe, miroir) ainsi que d'autres encore n'ayant pas d'équivalent.

### 5.3.2 La palette virtuelle

La palette virtuelle est un équipement d'interaction composé d'un manche surmonté d'une surface plane transparente (voir figure 1.10). Ses déplacements sont repérés dans l'espace à l'aide d'un capteur de position sur le manche. Coquillart et Wesche [31] présentent ce périphérique d'interaction en 1999 ainsi qu'une technique de contrôle d'application appelée *Virtual Remote Control Palette* (voir figures 1.10 et 5.2). La palette est transformée en un dispositif de contrôle distant, comme une télécommande pour l'application. Des éléments graphiques 2D sont affichés sur la partie transparente de la palette pour constituer des menus contrôlant l'application. La palette est tenue en main non dominante et un stylus en main dominante permet d'actionner les menus.

Schmalstieg et al. [104] développent la même année un périphérique similaire nommé *pad* (ou bloc-notes en français) composé uniquement d'une plaque de verre transparente, sans manche, repérée dans l'espace par un capteur (voir figure 5.3).

Les auteurs proposent pour le *pad* une technique de contrôle d'application fondée sur



FIG. 5.3 – Le pad de Schmalstieg et al. [104]

l'affichage de menus 2D sur la surface transparente, identique à celle de Coquillart et Wesche. Ils décrivent également toute une gamme de nouvelles fonctionnalités :

- *Window Tools* (outils fenêtre) : La surface du *pad* sert de fenêtre sur le monde, à travers laquelle différents filtres peuvent être appliqués. Des techniques de manipulation 2D, avec le stylus sélectionnant sur la surface du *pad*, permettent d'agir "à travers" cette fenêtre sur un monde 3D altéré par le filtre. L'univers 3D est ainsi divisé en deux espaces de travail. Une deuxième application des *Window Tools*, proposée par Schmalstieg et al., crée des "photographies" de la scène selon des angles déterminés par l'orientation du *pad*. Elles agissent comme des points de vue supplémentaires sur la scène, sous forme de fenêtres qui peuvent être laissées flottantes à des endroits quelconques de la scène. L'utilisateur dispose alors de plusieurs vues d'un même objet, comme c'est le cas dans les applications courantes de CAD.
- *Through-the-Plane Tools* (outils à-travers-le-plan) : Schmalstieg et al. présentent deux techniques de sélection d'objets utilisant le *pad*. La première sélectionne les objets les plus proches du centre du *pad*. La seconde permet de dessiner au stylus un contour fermé sur le *pad* pour entourer comme au lasso les objets à sélectionner.
- *Outils de manipulation volumétrique* : Une nouvelle métaphore est proposée pour capturer des objets en déplaçant le *pad* dans la scène. L'utilisateur s'en sert comme d'un filet à papillon. L'image des objets traversés par la surface du *pad* reste affichée sur le *pad* lui-même, pour sélection ou manipulation ultérieures.

### 5.3.3 Le *Magic Mirror*

#### 5.3.3.1 Le miroir comme outil d'exploration

L'homme a inventé ou découvert au cours de son histoire de nombreux outils pour améliorer ou étendre ses capacités visuelles au delà de leurs capacités naturelles. Ces outils varient du plus simple au plus complexe : loupe, jumelles, télescope, microscope, endoscope, périscope, etc. Un des plus vieux, des plus simples et des plus connus est probablement le miroir. En tant qu'outil d'exploration de scènes 3D, le miroir présente un certain nombre d'avantages :

- La réflexion du miroir est facile à simuler.
- Le miroir est connu de tous. Son concept est familier et ne nécessite pas une phase d'apprentissage.
- Le miroir propose un approche exocentrique [59]. C'est un objet extérieur à l'utilisateur dont la manipulation est facilitée parce qu'elle est relative à son cadre de référence et peut exploiter le sens proprioceptif.
- Toute la complexité de l'interaction avec l'outil se résume à la tâche de positionnement et d'orientation d'une raquette tenue en main. Le comportement d'un miroir manipulé à la main est connu. Les utilisateurs ont déjà pu s'entraîner, au cours de diverses activités de leur vie (régler un rétroviseur, s'observer sous différents angles).
- Le miroir fournit à l'utilisateur un second point de vue sur la scène, tout en s'intégrant naturellement à l'intérieur de celle-ci.

En contrepartie, le miroir peut être déroutant par l'inversion de la vue et n'est pas adapté à toutes les tâches d'exploration. Deux scénarios peuvent s'avérer déroutants :

- Se guider dans l'espace à l'aide de la seule vue du miroir (i.e. sans se servir du cadre de référence de la vue principale).
- Utiliser le miroir lorsque l'utilisateur et le miroir sont tous les deux mobiles simultanément.

Le miroir est un outil trouvant son identité comme vue d'appoint, complémentant la vue principale, de préférence fixe pendant la manipulation du miroir, et n'est pas adapté à un fonctionnement isolé. Le miroir est a contrario très profitable pour se substituer à

de courts déplacements, pour explorer son environnement en restant immobile.

### 5.3.3.2 Version de base du *Magic Mirror*

Le *Magic Mirror* est une extension de la MagicSphere [28] et de la catégorie plate des Magic Lenses [6]. C'est une interface utilisateur qui dans un premier temps reproduit la réflexion d'un vrai miroir sur la surface plate d'une palette tenue en main (voir figure 5.4).

Le *Magic Mirror* est un outil d'exploration destiné aux univers virtuels immersifs. Son élaboration a cependant débuté sur station de travail, en remplaçant l'interaction manuelle par un contrôle indirect à la souris de l'avatar de la palette virtuelle. Son utilisation s'est révélée également profitable sur ce support, pour des raisons parfois différentes, aussi nous présentons ici deux versions du *Magic Mirror* :

- Dans sa version pour station de travail, le *Magic Mirror* est une surface réflechissante carrée, entourée d'un cadre et d'un manche 3D formant l'avatar d'une palette, contrôlée dans ses déplacements par des périphériques standards : souris et clavier.
- Dans sa version pour environnement virtuel, le *Magic Mirror* est une surface réflechissante carrée affichée sur la vitre transparente d'une palette physique, dont la position repérée dans l'espace par un capteur, sert à positionner l'affichage.

Il peut être noté qu'une version hybride est également réalisable en l'absence de palette physique disponible. La palette peut être simulée dans l'environnement virtuel en affichant l'avatar virtuel d'une palette et la surface réflechissante dans la scène, et en attachant les mouvements de cet objet 3D à la position et l'orientation d'un capteur (seul ou faisant partie d'un équipement plus complexe déjà présent, comme le stylus).

L'accessibilité et la disponibilité de la palette sont les deux premiers points à considérer pour l'utilisation du *Magic Mirror* comme outil d'exploration. Dans le cas d'une manipulation par l'intmédiaire d'un *prop* physique, la disponibilité du *Magic Mirror* est directement liée à l'accessibilité de l'outil physique, selon que l'utilisateur l'ait constamment en main ou puisse le poser près de lui.

Sur station de travail, la palette est représentée par un avatar, c'est à dire un objet indépendant. Elle ne doit pas se perdre même lorsque l'utilisateur se déplace dans la scène

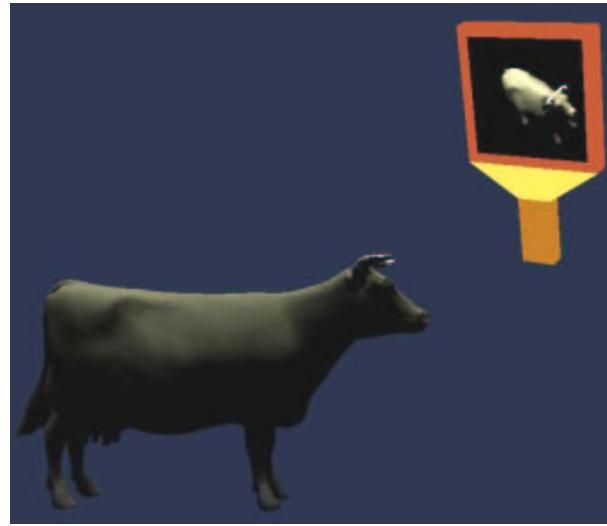


FIG. 5.4 – Le *Magic Mirror*.

virtuelle. En tant qu'outil d'interaction, elle doit être à tout moment à portée de main. Cet objectif est accompli en attachant l'avatar de la palette au repère de la caméra. La palette devient solidaire des mouvements de la caméra et reste immobile dans son repère. Ainsi, elle ne peut plus quitter le champ de vision pendant la navigation.

#### 5.3.3.3 Extensions du *Magic Mirror*

Afin d'étendre les possibilités d'exploration offertes par le simple miroir, des fonctionnalités supplémentaires, réalistes ou impossibles dans le monde réel, sont proposées et réalisées pour le *Magic Mirror*.

**5.3.3.3.1 Taille** Pour faire face à différentes tailles de modèles à explorer, il est nécessaire de pouvoir faire varier la taille du miroir lui même (et donc celle de la palette) dans le monde. A cette fin, il est possible d'ajuster interactivement les dimensions de la palette virtuelle dans la scène. La palette physique elle, ne pouvant changer, c'est le monde virtuel qui doit être réduit ou augmenté en conséquence.

**5.3.3.3.2 Ouverture** Une des premières fonctionnalités que nous avons ajoutée au miroir est la possibilité de changer interactivement l'angle de réflexion pour capturer soit une plus grande, soit une plus petite portion de l'espace au sein de l'image réfléchie. Le

*Magic Mirror* devient alors un outil capable de produire une image réfléchie montrant finement un détail lointain ou capturant au contraire un très large champ de vision.

**5.3.3.3.3 Grossissement** Pour donner à l'utilisateur plus de confort quant à l'observation d'un détail capturé par la réflexion du miroir, nous donnons la possibilité de changer interactivement le facteur d'agrandissement de l'image réfléchie. Le grossissement de l'image est cependant une simple opération 2D d'étirement, effectuée sur l'image réfléchie originelle, dont il ne change pas la résolution initiale.

**5.3.3.3.4 Inversion de l'image** Une faiblesse bien connue du miroir comme outil d'exploration est l'inversion de l'image réfléchie. Dans le cas d'éléments graphiques contenant des informations alphanumériques notamment, comme des lettres écrites sur la surface d'un objet 3D, le décryptage de l'information portée par le miroir devient très délicat. C'est pourquoi nous ajoutons au *Magic Mirror* la possibilité de retourner l'image réfléchie selon un axe médian vertical. Ce mode supplémentaire peut être activé ou désactivé sur commande. Il n'est en effet pas souhaitable de fonctionner en permanence avec l'inversion d'image à cause des besoins de lecture, ni avec l'image rétablie, car un miroir reflétant une image non renversée est hautement déroutant à manier.

**5.3.3.3.5 Dégagement de la vue du miroir** Lors de la manipulation d'un outil d'exploration de scène comme le *Magic Mirror* il est fondamental que la surface réfléchissante reste visible à tout instant. Une fonctionnalité cruciale du *Magic Mirror* permet de rendre invisible tout objet ou partie d'objet occultant la vue du miroir. Plus précisément, le *Magic Mirror* efface de la scène tout ce qui pénètre une pyramide ayant pour sommet la position de la caméra principale (l'œil gauche puis droit lors du calcul des vues stéréoscopiques) et pour base la surface carrée réfléchissante du miroir. Cette pyramide sera appelée dans la suite *frustum de dégagement du miroir* et est représentée sur la figure 5.5.

Cette fonctionnalité est la source de nombreux avantages. Tout d'abord cela assure que l'outil miroir est toujours visible et fonctionnel à tout instant, quelque soit sa position dans la scène et son occlusion par des objets situés entre la palette et l'utilisateur. Une utilisation dérivée de ce principe suggère d'utiliser le miroir pour explorer des zones préalablement hors du champ de vision de l'utilisateur : derrière un obstacle ou à l'intérieur

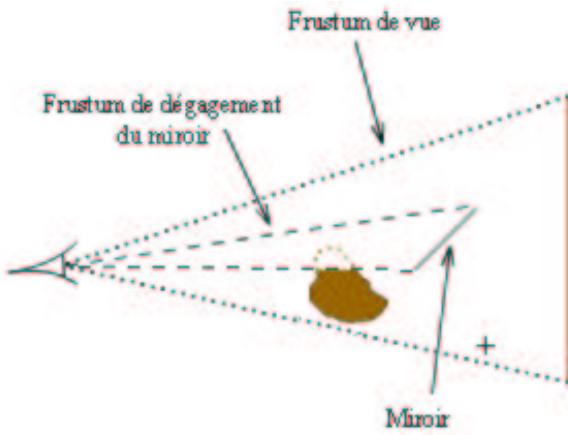


FIG. 5.5 – Frustum de dégagement du miroir et frustum de vue de la scène

de structures closes.

Il n'est pas toujours ais   ou souhaitable de naviguer jusqu'   un point d'int  ret dans une sc  ne, juste pour examiner ce qu'il rec  le. Le miroir peut ainsi servir    examiner le contenu d'un meuble    tiroirs, sans avoir    p  n  trer et naviguer difficilement    l'int  rieur de chaque   tage de tiroir, o   la visibilit   est grandement r  duite par l'  troitesse et l'obscurit   de ces zones herm  tiquement closes. La palette peut   tre amen  e ais  ment    l'int  rieur d'un tel meuble, l'utilisateur ayant    la fois une vue ext  rieure, immobile et d  gag  e de la surface r  f  chie et la perception de son bras, pour se guider dans la manipulation.

**5.3.3.3.6 Basculement vers la vue du miroir** Le miroir offre un second point de vue sur la sc  ne, disponible pour l'exploration. Il est obtenu simplement, sans se d  placer, par manipulation de la palette, dans un contexte exocentrique. Afin de mieux exploiter les b  n  fices de ce point de vue additionnel, nous donnons la possibilit      l'utilisateur de basculer son propre point de vue subjectif sur celui du miroir. Le miroir peut ainsi servir    trouver un point de vue int  ressant, et dans un second temps l'utilisateur r  cup  re ce point de vue comme vue principale, en se "t  l  portant"    un nouvel endroit de l'espace, correspondant    l'ancien du point de vue du miroir. Bien entendu, le *Magic Mirror* suit l'utilisateur dans ce "bond" et reste    la m  me position relative.

**5.3.3.3.7 Application de filtres** La r  flexion du miroir fournit un point de vue additionnel sur la sc  ne, dont l'orientation est guid  e par la m  taphore du miroir. Nous

remarquons qu'il est possible de diviser l'espace de travail en deux ensembles disjoints : le monde virtuel et le monde virtuel réfléchi, c'est à dire un monde doublement virtuel, dont on ne voit que la réflexion, et qui se superpose au monde virtuel précédent. Dans le cadre d'un miroir naturel, les deux mondes virtuels sont identiques (mais inversés). Il est cependant possible d'appliquer toutes les techniques de filtres et de générer un monde réfléchi différent.

Prenons pour exemple la luminosité de la scène. Puisque le miroir est visible en toutes circonstances et peut plonger à l'intérieur d'objets creux et sombres, il peut être utile de modifier la luminosité de l'image réfléchie. L'image elle-même n'est pas rendue plus lumineuse, c'est ici la scène montrée par la réflexion du miroir qui devient mieux éclairée. La scène vue par le regard direct de l'utilisateur ne change pas.

Le *Magic Mirror* peut donc accueillir dans la métaphore du miroir toutes les techniques de filtres possibles, en réfléchissant une vue d'un monde virtuel secondaire.

#### 5.3.3.4 Avantages pour l'exploration

Les avantages du *Magic Mirror* sont multiples, et peuvent s'étendre à de nombreux scénarios par la combinaison de ses fonctionnalités. Nous avons identifié cinq tâches caractéristiques qui selon notre expérience, bénéficient de l'utilisation du *Magic Mirror*.

**5.3.3.4.1 Exploration multi-échelle** L'exploration multi-échelle, c'est à dire la faculté de disposer rapidement de plusieurs vues d'une même scène ou d'un même objet à différentes échelles est très importante. L'exploration multi-échelle requiert d'être capable de basculer très rapidement d'un niveau d'observation à un autre (par exemple, basculer d'une vue microscopique à une vue macroscopique et inversement) ou encore plus commodément, de disposer de ces différentes vues simultanément.

C'est une tâche pratiquement impossible pour les techniques de navigation standard ayant un seul point de vue. Une solution consiste à se rapprocher ou s'éloigner rapidement de l'objet ou du détail considéré [84]. Cependant il est quasiment impossible une fois le point de vue rapproché, de revenir exactement, rapidement et avec facilité à sa position de départ.

Comme nous l'avons vu dans l'état de l'art, certains auteurs ont proposé des solutions avec de multiples vues, mais ces vues supplémentaires servent principalement à faciliter la navigation dans la vue principale. Il s'agit en général d'une vue globale et éloignée de la base de données. Un des défauts de cette technique est d'une part que l'intégration de la seconde vue n'est pas souvent naturelle. D'autre part, une seule de ces deux vues est en général pleinement modifiable.

La vue supplémentaire du *Magic Mirror* est au contraire intégrée naturellement dans l'environnement de travail par le biais de la métaphore du miroir. La seconde vue est totalement contrôlable par l'utilisateur pour montrer aussi bien une vue globale qu'un détail, grâce au contrôle interactif de l'ouverture de la vue du miroir.

**5.3.3.4.2 Exploration d'un objet sous de multiples angles** Tourner autour d'un objet afin de l'observer sous différents angles de vue est une tâche généralement malaisée pour la plupart des techniques de navigation actuelles. Elles se prêtent plus naturellement à une logique de déplacement vers l'avant dans une direction contrôlée, qu'à une succession de manœuvres pour tourner autour d'un objet fixe. Déplacer le point de vue principal autour d'un objet immobile demande en effet à ces techniques une succession fastidieuse de translations et de réorientations. A l'opposé, observer un objet immobile sous de multiples angles de vue est une opération beaucoup plus naturelle et rapide à l'aide d'un miroir. Cette tâche ressemble à celle consistant à tourner un miroir autour de soi ou de son visage pour l'observer sous différents angles. Pour cette tâche, il est nécessaire là aussi de combiner déplacement et réorientation, mais cette opération est effectuée sur un objet extérieur tenu en main, profitant ainsi des 6 degrés de liberté de la main, et de toute l'expérience manuelle de l'utilisateur.

**5.3.3.4.3 Explorer à travers une surface opaque** Il est possible d'explorer au delà d'une surface opaque par l'utilisation de plans de coupe. Leur usage n'est pas idéal cependant, car ils coupent l'espace en deux et élaguent une moitié de son contenu. L'obstacle est certes retiré de la scène, permettant de voir ce qu'il cachait, mais de nombreux éléments disparaissent en même temps. A cet égard, le dégagement de la vue d'un miroir constitue une meilleure approche. Elle permet de retirer de la vue principale les éléments qui masquent la surface du miroir. Or cette surface capture une vue au delà de l'obstacle, pour laquelle la scène n'est pas altérée. Le *Magic Mirror* est de plus très flexible car il pro-

pose de multiples vues (sous différents angles contrôlables) de la scène derrière l'obstacle opaque, sans bouger le point de vue, par simple rotation de la palette sur son centre.

**5.3.3.4.4 Explorer l'intérieur de petits objets** Explorer l'intérieur de petits objets est assez délicat avec des techniques de navigation, car elles obligent le point de vue à entrer dans l'objet. Si l'objet est trop petit, l'utilisateur ne dispose plus alors du recul nécessaire pour observer convenablement l'environnement intérieur de l'objet. Dans le cas du *Magic Mirror*, le point de vue de l'utilisateur reste à l'extérieur de l'objet considéré. Le *Magic Mirror* est tenu en main et pénètre l'objet tout en tronquant les éléments masquant sa surface. L'utilisateur peut manipuler la position et l'orientation de la surface du miroir pour observer l'intérieur de l'objet avec tout le recul nécessaire offert par sa position éloignée.

**5.3.3.4.5 Retrouver des objets hors du champ de vision** Dans leur utilisation courante, un écran d'ordinateur tout comme le Plan de Travail Virtuel proposent une vue partielle sur un univers 3D. A un instant donné, une grande partie du monde virtuel n'est pas directement visible. Pour explorer tout son environnement immédiat, un utilisateur doit généralement se déplacer ou faire pivoter sa représentation dans le monde, pour observer l'univers sous d'autres angles (l'écran et le Plan de Travail Virtuel sont physiquement immobiles bien sur, seul l'affichage s'accorde à cette rotation du point de vue). Dans ce contexte où moins de 50% de l'univers est visible, il n'est pas rare de "perdre" des objets 3D hors de son champ de vision, ou de simplement vouloir retrouver des objets que l'on sait proches mais non visibles. Les modèles sont souvent affichés flottant dans un espace vide, sans repère. Naviguer pour retrouver un objet dans l'espace est souvent déroutant, car l'absence de repère ne permet pas de ressentir l'amplitude de ses mouvements ou de ses rotations. L'approche exocentrique du miroir est donc ici particulièrement profitable, car elle permet de réduire le problème de l'exploration à la manipulation d'un objet externe, visible, présent dans un cadre de référence défini : celui du corps de l'utilisateur. Le facteur d'agrandissement interactif de l'angle de vision du miroir est pleinement exploité pour capturer une large portion de l'espace normalement invisible à l'utilisateur. Des lors, de très faibles rotations de la palette sont nécessaires pour balayer l'ensemble de la zone cachée. L'utilisateur n'est pas contraint de modifier son point de vue, et conserve toute son orientation spatiale pendant la recherche. Enfin s'il le désire, il peut basculer sa vue

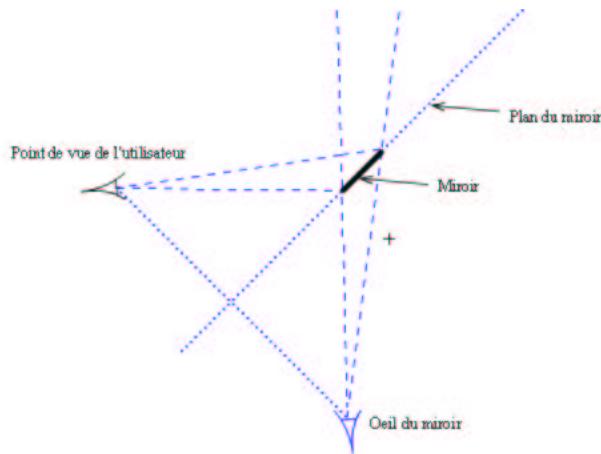


FIG. 5.6 – Calcul de la vue du miroir

sur celle du miroir et ainsi, en une seule opération, retrouver l'objet devant lui.

### 5.3.3.5 Réalisation et détails techniques

La réflexion du miroir peut être simulée simplement. L'image réfléchie correspond à la vue d'un œil fictif, symétrique de l'œil réel de l'autre côté du plan du miroir. Ce second point de vue sur la scène s'appuie sur les bords du miroir pour son frustum de vue comme le montre la figure 5.6.

L'image perçue par le point de vue de cet *œil du miroir* est calculée en premier. Pour ce calcul la palette est retirée de la scène, pour l'empêcher de masquer la vue s'étendant au delà de son cadre. Cette image sert ensuite de texture à la face carrée de la palette représentant le miroir. La scène réelle peut maintenant être calculée.

Deux vues sont ainsi calculées à chaque instant, d'abord celle du miroir, puis celle incluant le miroir et son image réfléchie. En réalisant ces opérations à chaque boucle graphique, l'image réfléchie s'adapte en temps réel aux mouvements du point de vue et de la palette elle-même. Il faut noter toutefois que cette opération nécessite une puissance graphique importante.

La position de l'œil du miroir sert également lors du basculement de la vue sur celle du miroir. Il suffit de récupérer toutes les coordonnées spatiales de l'œil du miroir et de les appliquer au point de vue de l'utilisateur.

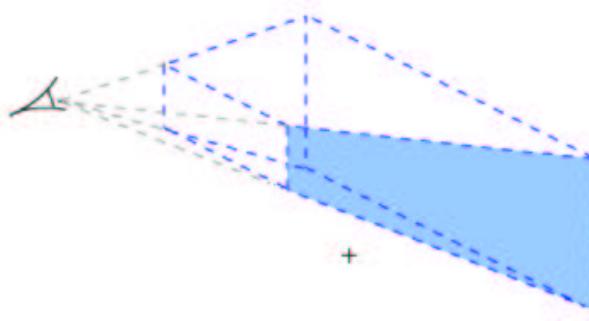


FIG. 5.7 – Le frustum de vue est représenté en traits bleus. Le plan bleu clair forme le bord droit du frustum. Lorsqu'il est activé, seul ce qui se situe à sa droite est dessiné. Le même procédé est suivi pour les plans gauche, haut et bas, avant et arrière.

La modification de l'angle de réflexion du miroir est simulée en déplaçant l'œil du miroir le long d'une droite passant par le symétrique de l'œil réel et le centre du miroir. Lorsque l'œil du miroir est rapproché de la surface, l'angle de capture s'ouvre et le miroir réfléchit une plus large portion du monde. Lorsque l'œil du miroir est éloigné de la surface, l'angle de capture se rétrécit et le miroir ne réfléchit plus qu'une faible portion du monde.

En modulant les paramètres d'éclairage naturel de la scène pour le calcul de la vue du miroir, il est aisément de produire une image réfléchie plus ou moins sombre, sans altérer la vue globale de la scène, et donc sans éclairer les autres objets de la scène qui ne le nécessitent pas.

L'inversion de la surface du miroir, pour permettre par exemple la lecture de texte, est trivialement réalisée. Il suffit d'inverser gauche et droite dans l'attribution des bords de la texture aux bords de la palette.

Le seul problème rencontré concerne la meilleure technique pour réaliser l'élagage des objets ou parties d'objets à l'intérieur du frustum de dégagement du miroir. Viega et al. [118] traitent ce problème en dessinant la vue en plusieurs passes. A chaque rendu graphique, un plan de coupe est utilisé pour dessiner successivement les demi-espaces droit, gauche, haut, bas, avant et arrière (voir figure 5.7). Ces plans de coupe forment les bords de la pyramide. Au dernier rendu, toute la scène est dessinée, sauf l'intérieur de la pyramide.

Pour éviter de dessiner la scène en de nombreuses passes, nous développons une mé-

thode plus efficace et plus simple. Nous dessinons tout d'abord la vue sans la présence du miroir, puis nous dessinons le miroir par dessus cette image, sans nous préoccuper des faces cachées. De cette façon, la surface réfléchissante apparaît toujours devant tout élément de la scène.

### 5.3.4 Magic Windows

Le concept du *Magic Mirror* peut être "retourné" pour obtenir très simplement une seconde technique d'exploration exploitant les résultats précédents. A la place d'une image réfléchie, la surface de la palette peut agir comme une loupe, autre objet bien connu du monde réel, offrant une vue déformée de l'univers virtuel. Par extension, la palette peut offrir un support physique à une multitude de filtres 3D, au delà du simple effet de grossissement propre à la loupe.

Une difficulté évoquée pour l'utilisation de filtres 3D dans les environnements virtuels concerne les moyens mis en œuvre pour contrôler la position et l'orientation des filtres dans l'univers. La palette est un outil idéal pour localiser des lentilles plates de Viega et al. [118] et offrir une composante physique manipulable et un retour tactile à l'utilisateur. La surface utile plate de la palette agit comme un *prop* matérialisant la lentille plate virtuelle, à la manière des *Voodoo Dolls* de Pierce [96], décrites dans le chapitre état de l'art. L'apparence de la plaque de verre de la palette rappelle celle d'une loupe ou plus généralement d'une lentille. Comme dans l'étude médicale de cerveaux humains de Hinckley [62] par une plaque tenue en main symbolisant un plan de coupe, la palette apporte un retour tactile facilitant le placement spatial de la lentille.

La palette utilisée comme support pour l'intégration de lentilles plates partage également un certain nombre d'avantages avec le Magic Mirror. Nous pouvons notamment citer :

- L'effet de lentille plate est facile à simuler.
- L'effet loupe est connu de tous. Son concept est familier et ne nécessite pas une phase d'apprentissage.
- La loupe propose un approche exocentrique [59]. C'est un objet extérieur à l'utilisateur dont la manipulation est facilitée parce qu'elle est relative à son cadre de

référence et peut exploiter le sens proprioceptif.

- Toute la complexité de l'interaction avec l'outil se résume à la tâche de positionnement et d'orientation d'une raquette tenue en main. Le comportement d'une loupe manipulée à la main est connu. Les utilisateurs ont déjà pu s'entrainer.
- La lentille s'intègre naturellement dans l'univers virtuel grâce au concept de la palette.

Les *Magic Windows* forment une gamme d'outils dédiés à l'exploration de scènes 3D, se fondant sur le principe des lentilles plates appliquées à la palette virtuelle. Ils bénéficient de certaines extensions du *Magic Mirror*, qui garantissent leur disponibilité permanente et étendent leur champ d'action. Une des fonctions primordiales du *Magic Mirror* permet de dégager la vue du miroir. En associant cette fonction à la technique des lentilles plates pour la palette virtuelle, les *Magic Windows* garantissent une disponibilité permanente de l'outil et une extension de ses capacités pour l'exploration au delà d'obstacles opaques ou à l'intérieur d'objets 3D. La seconde extension reprise du *Magic Mirror* est l'ouverture réglable du champ de vision de la lentille plate. Les lentilles plates ouvrent un point de vue différent sur la scène. Cette fonctionnalité donne aux *Magic Windows* un meilleur contrôle de cette fenêtre, en permettant à l'utilisateur de choisir l'étendue du panorama capturé, du plus petit détail jusqu'à un demi-espace entier.

### 5.3.5 Evaluations informelles

Le *Magic Mirror* a été testé de manière informelle dans sa première version sur station de travail pour un petit nombre d'utilisateurs. L'outil permet une exploration approfondie de toute zone de l'espace de travail, sans restriction. Tout objet peut être observé sous n'importe quel angle, quelque soit sa taille, son éclairement ou son emplacement dans la scène, devant, derrière ou à l'intérieur d'un autre objet. La manipulation sur station de travail est réalisée à la souris, périphérique à deux dimensions et les déplacements de la palette doivent être décomposés. L'interaction est de ce fait jugée parfois trop lente. Ce défaut n'apparaît pas cependant en environnement immersif où la palette est directement tenue en main et aisément déplaçable et manipulable.

## 5.4 Conclusions et perspectives

Les filtres sont des outils puissants et versatiles dans les environnements 2D. L'intégration de filtres dans un univers 3D rencontre deux difficultés pour lesquelles nous avons proposées des solutions. Il faut d'une part déterminer l'apparence des filtres 3D, la portion de l'espace qu'ils occupent et celles sur laquelle ils s'appliquent. Cette apparence jouera sur la façon dont ils se combinent les uns avec les autres. D'autre part, une méthode pour contrôler leur position et leur orientation dans l'espace doit être également mise en œuvre. Pour résoudre ces deux points, nous présentons deux métaphores exploitant la palette virtuelle de Coquillart et Wesche [31] : le miroir et la loupe. S'appuyant sur la double métaphore de l'outil et de la palette, cette méthode d'exploration de scènes 3D est intuitive et facile à manipuler.

Le miroir est un des outils communs du monde réel, manipulé à l'aide d'une palette. Il étend la vue de l'utilisateur au delà de son champ de vision naturel. Nous avons simulé l'outil miroir, à la fois sur station de travail, à l'aide d'un avatar virtuel de la palette, et avec une palette physique dans un environnement virtuel. La vue additionnelle qu'offre la palette à l'utilisateur est ainsi intégrée naturellement dans le monde virtuel grâce à la métaphore du miroir. Elle est entièrement contrôlable par l'utilisateur qui dirige l'orientation de la palette. L'outil ainsi créé est baptisé *Magic Mirror*.

Le miroir peut être déconcertant par sa vue inversée lorsque l'utilisateur est lui-même en déplacement ou si il n'a accès qu'à la vue inversée du miroir pour se guider. Dans le cadre d'utilisation normale, où il est immobile et se sert du miroir pour explorer son environnement, l'utilisateur bénéficie de son expérience dans le monde réel pour la manipulation du *Magic Mirror*.

Des extensions ont été apportées au *Magic Mirror* pour accroître ses avantages pour l'exploration de scènes 3D : taille et grossissement variable du miroir, inversion de l'image pour permettre la lecture de textes réfléchis, basculement de la vue de l'utilisateur vers celle du miroir. D'autres extensions se concentrent spécifiquement sur l'assurance de la visibilité du miroir en toutes circonstances, pour garantir son intérêt comme outil d'exploration de scènes 3D : la vue du miroir est dégagée des objets entrant dans la pyramide de vue de l'utilisateur et empêchant de voir la surface. Il peut dès lors employer le miroir pour explorer la scène au delà d'objets opaques ou à l'intérieur d'objets creux. Une

avant-dernière extension permet de maîtriser les dimensions de la zone réfléchie par l'outil miroir. Comme pour un appareil photographique, l'ouverture de la caméra est réglable. Pour le miroir cela veut dire qu'il est possible de modifier interactivement l'angle de réflexion et de jouer ainsi sur les dimensions de l'espace capturé par la réflexion, du plus petit détail à un demi-espace entier. Enfin, la palette sert de support tactile pour la localisation spatiale et le contrôle des filtres. Le *Magic Mirror* intègre dans sa métaphore la technique des filtres en proposant une réflexion d'un monde virtuel différent du monde virtuel qui entoure la palette.

De même que la métaphore du miroir, la palette peut abriter la métaphore de la loupe. La surface transparente agit comme une lentille, altérant la vue du monde virtuel. Les filtres 3D choisis s'appuient sur le paradigme des lentilles plates de Viega et al. [118]. La surface de verre de la palette sert de *prop* représentant la surface de la lentille plate virtuelle avec laquelle elle se superpose. Cette intégration des lentilles plates sur la palette virtuelle à la manière d'une loupe tenue en main, est baptisée *Magic Windows*. Elle bénéficie des extensions du *Magic Mirror* augmentant le champ d'action de la palette : réglage de l'ouverture et dégagement de la vue.

Nous avons vu comment la palette virtuelle peut abriter deux métaphores d'outils pour l'exploration visuelle de scènes 3D, l'intégration de filtres et d'extensions augmentant leurs champs d'action et d'utilisation. De nombreuses fonctionnalités sont ainsi condensées en un même équipement. Il est toujours important en réalité virtuelle de ne pas multiplier les équipements à disposition pour interagir avec la scène. Une des faiblesses de la palette cependant, concomitante à sa nature de *prop*, est qu'elle n'est positionnable qu'à portée de mains (bien qu'elle puisse donner une vue d'objets distants). Comme pour toutes les techniques de sélection/manipulation, la manipulation à distance demande une considération particulière. Des techniques hybrides, exploitant les résultats de l'état de l'art pourraient être envisagées, comme par exemple un *go-go mirror*, où des avatars de la main et de la palette sont contrôlés par la technique *Go-Go* [99] de Poupyrev, avec un capteur attaché à la main.

# Chapitre 6

## Minifly : une plate-forme logicielle pour environnement virtuel

### 6.1 Présentation

Du fait de la jeunesse des configurations d'environnement virtuel, il n'existe pas encore de standard logiciel pour le développement d'application en réalité virtuelle, comme il en existe sur les stations de travail. La diversité des équipements de visualisation et d'interaction a été propice à une diversification des solutions logicielles. Des configurations de réalité virtuelle proches, comme les configurations à base de projection sur grands écrans, sont caractérisées par un certain nombre de paramètres d'affichage distincts : la disposition et le nombre des écrans, et leur taille. Les applications doivent s'adapter à ces différences, proposer pour l'interaction une gestion spécifique des périphériques selon leur technologie, un modèle de gestion logicielle des outils d'interaction et un système de contrôle d'application.

Pour les besoins de nos applications, nous avons donc développé une plate-forme commune à toutes nos applications, fondée sur la librairie graphique Performer<sup>TM</sup> et gérant tous ces aspects, ainsi qu'un noyau logiciel pour le développement et l'ajout de nouveaux outils et paradigmes d'interaction.

Nous présentons dans ce chapitre Minifly, cette plate-forme logicielle pour environne-

---

ment virtuel, que nous avons développée dans le cadre de la thèse. Minifly se propose d'offrir une base commune au développement d'applications immersives en couplant deux modules indépendants l'affichage et la gestion des outils d'interaction.

## 6.2 Affichage

L'affichage de Minifly s'appuie sur la bibliothèque graphique Performer<sup>TM</sup>, qui présente des fonctionnalités avancées pour la gestion de vues multiples stéréoscopiques synchronisées exploitant plusieurs cartes graphiques. Un programme de démonstration des possibilités graphiques de cette bibliothèque existe sous le nom de Perfly. Ce programme permet sur une station de travail standard de charger des modèles 3D provenant de diverses applications de modélisation et enregistrés dans de nombreux formats d'enregistrement populaires. Il permet de les visualiser à l'écran et de les manipuler à l'aide d'opérations basiques à la souris et au clavier. Un menu permet le réglage d'un grand nombre d'options de visualisation, et d'optimisation graphique. La partie affichage de Minifly est construite sur le programme Perfly, en l'étendant sur plusieurs points, pour fonctionner aussi bien sur station que sur notre configuration de Plan de Travail Virtuel :

- Affichage sur un ou deux écrans.
- Affichage stéréoscopique. Chaque écran affiche deux images successives calculées pour deux points de vue proches, à très haute fréquence.
- Affichage pour un point de vue mobile dans l'espace. Lorsque les mouvements de la tête sont enregistrés, des capteurs sont interrogés périodiquement pour connaître la position courante du centre des deux yeux. Notre configuration propose notamment un gestionnaire pour des capteurs électromagnétiques.

Minifly peut fonctionner indifféremment avec un ou deux écrans, en vision mono- ou stéréoscopique, avec suivi ou non des mouvements de la tête et permettre de charger et manipuler des modèles 3D. Cette flexibilité permet notamment de faciliter la capture d'images à des fins d'illustration en basculant l'application en mode non stéréoscopique.

## 6.3 Gestion des outils d'interaction

Le second module de Minifly, gère la création, l'appel, le renvoi et la destruction des outils d'interaction. Il est conçu avec un souci d'indépendance vis à vis de la partie graphique. Pour ces outils, cette partie graphique est isolable facilement pour s'adapter à d'autres supports. Le module outil peut ainsi être facilement exporté sur d'autres systèmes.

Le premier aspect de la gestion des outils d'interaction en environnement virtuel concerne l'enregistrement des données de position. Deux gestionnaires (ou *drivers* en anglais) pour capteurs électromagnétiques ont été créés. Le premier gestionnaire permet simplement de récupérer les valeurs de position et d'orientation des capteurs à six degrés de liberté. Le second gestionnaire enregistre toutes les données venant des capteurs, vers un fichier de stockage. Selon le gestionnaire adjoint au programme, il est possible au choix de conserver ou non une trace des données des capteurs, par exemple le temps d'une expérience, pour enregistrer les performances d'un sujet.

Les techniques d'interaction en environnement virtuel se résument souvent par la manipulation d'une gamme d'outils appelés selon les besoins de l'utilisateur et associés à l'une ou l'autre main. Ces outils sont représentés dans Minifly comme des composants logiciels autonomes. Un outil, au sens de Minifly, se compose de cinq éléments :

- Un ensemble de fonctions ou règles de comportement. Elles décrivent toutes les actions que peut accomplir l'outil.
- Une apparence graphique (optionnelle). Pour les outils faisant appel à des éléments graphiques 3D dans la scène virtuelle.
- Une partie physique (optionnelle). Pour les outils composés tout ou partie d'un élément physique dans le monde réel, autre que le capteur de position, et pouvant générer des signaux additionnels comme des boutons.
- Un retour haptique (optionnel). Lorsque l'outil comprend un dispositif de retour tactile ou d'effort, qu'il doit contrôler.
- Un retour sonore (optionnel). Lorsque l'outil comprend un dispositif sonore, qu'il doit contrôler.

Minifly fournit un modèle général à la conception d'outils, permettant d'intégrer l'en-

semble des caractéristiques précédentes. Il suit une approche objet permettant la réutilisation de caractéristiques communes, et facilite le développement de nouveaux outils ou l'extension d'outils préexistants.

Un outil est plus ou moins spécialisé selon la tâche qu'il doit réaliser. Nous avons créé un modèle commun d'outil, dont chaque outil futur hérite, lui conférant ainsi un ensemble de caractéristiques primordiales. Nous distinguons trois catégories de comportement : exécution d'un traitement périodique, production d'événements ponctuels et réaction à la réception d'événements.

L'exécution d'un traitement périodique permet de définir le comportement de base d'un outil, ainsi que de remplir toutes les tâches de routine nécessaires, comme par exemple, repositionner un élément graphique à chaque instant à la hauteur d'un capteur. Dans ce modèle, la technique du rayon virtuel est représentée par un outil "rayon". L'outil est composé d'une partie physique, le stylus, qui apporte le signal du bouton et les données du capteur de position, et d'une partie graphique, le rayon, à repositionner à chaque instant dans le prolongement du stylus. Le comportement de base de l'outil "rayon" est de vérifier à chaque instant les intersections du rayon graphique avec les éléments de la scène, et de les rendre solidaires du rayon lorsque le bouton est enfoncé.

La création, l'envoi et la réception d'événements permet de définir un modèle de communication entre les outils, et de lier leurs comportements, voire de créer des réactions en chaîne. A titre d'exemple, nous pouvons citer le menu  $C^3$ , intégré dans ce modèle. Il produit des événements lorsque son pointeur sphère entre dans une case et y reste un certain laps de temps. Il envoie un événement à destination d'un second outil "étiquette de texte", qui sait afficher des textes sur une étiquette plate, frottant dans l'espace et tournée vers le regard de l'utilisateur. Cet événement contient les lettres à afficher, ainsi que la position dans l'espace de l'étiquette de texte. Cette chaîne complète recrée le mécanisme des info-bulles. L'info-bulle est volontairement découpée du menu  $C^3$  pour mettre en œuvre le principe de modularité, favorisant la réutilisation des modules. Les étiquettes de texte sont notamment réemployées pour réagir à des événements d'affichage de lettres pour l'outil QuikWrite VR.

Minifly prend en charge d'autre part les mécanismes d'appel, d'utilisation et de renvoi de ces outils, en mimant les scénarios d'exploitation classiques. Les outils peuvent être attachés à deux points d'ancre logiciel, représentant les mains. A la fréquence de

l'affichage, les outils attachés exécutent leur traitements périodiques : positionnement ou altération des éléments graphiques, envoi d'événement, envoi d'une commande haptique ou sonore, etc. Plusieurs outils peuvent être attachés à chaque point d'ancrage. Lorsqu'ils ne sont plus souhaités, les outils sont détachés des points d'ancrage.

Le menu  $C^3$  est un outil attaché sur le point d'ancrage gauche (pour un droitier), pour représenter le fait qu'il est manipulé en main non dominante. En tant qu'outil, ses fonctions gèrent son affichage conditionnel (l'utilisateur doit le regarder), le déplacement de la sphère, la sélection d'une option de menu et l'exécution d'une fonction de l'application. En tant que technique de contrôle d'application, ses fonctions prennent en charge également l'appel des autres outils et leur attachement à un des deux points d'ancrage. Le  $C^3$  remplit ainsi son rôle de contrôle des techniques d'interaction.

## 6.4 Applications

Le logiciel minifly est utilisé comme plate-forme de développement par l'équipe i3D de l'INRIA sur un Plan de Travail Virtuel. La plateforme et ses différents outils d'interaction ont servi au développement d'applications diverses, parmi lesquelles : Fracfly, un programme de visualisation et d'exploration de formes fractales, un programme de modélisation à l'aide d'un curseur métrique [112] et un programme de modélisation et d'édition de surfaces implicites.



# Chapitre 7

## Conclusion et perspectives

### 7.1 Conclusion

Dans ce mémoire, nous avons tout d'abord présenté le cadre général de la réalité virtuelle, telle qu'elle est perçue par les différents acteurs de ce milieu et tenté d'en donner une définition fédératrice. Les configurations de réalité virtuelle les plus communes ont été exposées, tout particulièrement le Plan de Travail Virtuel, sur lequel ont été effectués les travaux de cette thèse.

Le **deuxième chapitre** s'est consacré à un état de l'art des différentes techniques d'interaction en réalité virtuelle. Nous avons présenté plusieurs taxonomies, aussi bien pour les équipements de l'interaction que pour les techniques d'interaction elles-mêmes. Trois grandes catégories de l'interaction en réalité virtuelle sont généralement reconnues comme tâches canoniques : la navigation, la sélection/manipulation et le contrôle d'application. Les deux premières catégories ont reçu à ce jour le plus d'intérêt et de développement de la part de la communauté des chercheurs.

Le contrôle d'application, bien que se situant comme pilier central de l'organisation de l'interaction au sein d'une application de réalité virtuelle, est encore peu approfondi. Les difficultés du contrôle d'application en environnement immersif sont liées au caractère immersif et direct de l'interaction et à l'ajout d'une dimension supplémentaire de sélection dans l'espace, encore mal maîtrisée.

---

Dans le **troisième chapitre**, nous avons livré notre intention de développer un système de raccourcis pour les environnements virtuels, c'est à dire une technique d'interaction contrôlant un jeu réduit de commandes mais dont l'objectif premier est la simplicité et la rapidité d'accès. De par sa définition notre objectif s'apparente à la technique des raccourcis claviers sur station de travail, dont la fonction est de fournir à l'utilisateur une technique simple et requérant une attention minimum pour appeler les commandes les plus usitées d'une application. Les raccourcis claviers se veulent complémentaires des menus linéaires donnant accès à la hiérarchie complète des fonctions d'une application, et n'ont pour vocation que d'accélérer le travail d'un utilisateur en lui facilitant l'accès aux fonctions les plus courantes. Ils évitent de détourner l'attention de l'utilisateur par une sélection fastidieuse dans un menu.

A cette fin, nous avons introduit une technique graphique similaire appelée *Marking Menus* [75], et exploitant l'ensemble des dimensions de l'espace (ici deux dimensions pour l'écran) pour la sélection. Les *Marking Menus* sont des menus *pop-up* circulaires, commandés par la souris, permettant deux niveaux d'expertise au sein du même paradigme : une sélection dans un menu graphique pour des utilisateurs novices et une sélection en tirant un trait avec le pointeur pour des utilisateurs expérimentés.

Nous avons ensuite montré comment étendre ce concept aux trois dimensions de l'espace, à travers une division de l'espace en 3x3x3 cubes. La structure cubique choisie présente les avantages de fournir un support mnémotechnique fort, tout en simplifiant la division complexe de l'espace à trois dimensions autour du pointeur. Elle permet aussi de s'aligner sur des directions simples de l'espace (gauche, droite, haut, bas, avant, arrière) que peut renforcer le sens proprioceptif de l'utilisateur lors de la manipulation, ainsi que des indices visuels puisque ces directions correspondent également aux cadres des écrans du Plan de Travail Virtuel.

Ce menu à accès rapide baptisé *Command and Control Cube* ou  $C^3$  se manipule à l'aide d'un équipement léger et peu envahissant : un simple capteur équipé d'un bouton. La simplicité et le faible encombrement du dispositif est un atout pour l'utilisation du  $C^3$  en environnement immersif. Le dispositif est accroché sur la main non-dominante. La main dominante reste ainsi volontairement libérée pour les tâches principales, propres à l'application elle-même. L'utilisateur peut conserver toute son attention sur le travail qu'il réalise, sans être contraint de briser sa concentration pour manipuler le système

de raccourci. De plus le  $C^3$  fonctionne, comme les *Marking Menus*, sur la base d'un paradigme d'interaction commun à une approche graphique et une approche gestuelle. Il peut ainsi être manipulé "en aveugle", c'est à dire sans retour visuel du menu graphique, par simple geste dans une direction de l'espace. L'utilisateur est ainsi en mesure d'appeler des commandes, sans détacher son regard, sa main dominante, ni son attention de la tâche principale. De plus, un système dual menu graphique et geste, mais construit sur une technique d'interaction unique, permet à l'utilisateur novice de progresser continuement vers la rapidité de sélection du mode aveugle de l'expert, et à l'utilisateur expert ayant un doute, de redescendre à tout instant de façon transparente au support graphique du mode novice.

Un soin tout particulier a également été porté aux considérations de visibilité et de placement des éléments graphiques, en relation notamment à l'obstruction de la vue ou des modèles de la scène. La disposition des menus graphiques est en effet particulièrement importante dans un univers combinant l'espace de manipulation à l'espace de visualisation et où les possibilités de placement sont très nombreuses du fait de la troisième dimension, et de l'immersion de l'utilisateur.

L'apparence générale du  $C^3$  est compacte, 26 cubes disposés tout autour d'un cube central. Les cases du  $C^3$  sont transparentes, pour ne jamais cacher d'éléments à la vue, avec un pointeur sphérique de petite taille pour la sélection. Le menu est invisible à l'état de repos, pour ne pas déranger. Il apparaît enfin à une position relative au corps de l'utilisateur : à une courte distance devant sa main. Ce positionnement à deux avantages principaux :

- Le  $C^3$  suit l'utilisateur dans ses déplacements. Quel que soit son parcours dans l'univers virtuel, le menu est toujours disponible et au même emplacement relatif.
- L'utilisateur choisit la meilleure position spatiale pour l'affichage. Puisqu'il est impossible de prévoir l'ensemble des scénarios d'utilisation, nous laissons le contrôle de son apparition dans les mains de l'utilisateur. La position de la main de l'utilisateur au moment de l'invocation du menu, détermine l'endroit de son apparition.

En tant que menu *pop-up* le  $C^3$  ne peut apparaître que sur invocation de l'utilisateur, i.e. si le bouton est pressé. Une condition supplémentaire d'apparition est ajoutée pour obtenir un comportement adaptatif entre les deux modes d'utilisation, novice (menu graphique) et expert (geste). Le  $C^3$  n'apparaît que si l'utilisateur porte son regard à l'endroit

attribué au  $C^3$  : devant sa main. Le capteur de tête permet de déterminer ainsi l'intention de l'utilisateur, selon qu'il regarde devant lui ou baisse la tête pour observer le menu.

L'ensemble de ces précautions permet d'assurer une obstruction visuelle minimum ainsi qu'une disponibilité permanente du  $C^3$ .

Des évaluations ont ensuite été menées auprès d'une population de 23 utilisateurs novices pour valider le principe du  $C^3$  et de la manipulation en aveugle en main non-dominante. L'étude des résultats a pu montrer des taux de réussite et des vitesses de sélection très encourageants pour un premier contact avec la configuration et le paradigme d'interaction. Les différences d'accessibilité des différentes cases ont également été révélées, notamment dans le contexte du Plan de Travail Virtuel par une certaine gène pour les cases du plan inférieur, trop proches du bras pour permettre un mouvement rapide et non retenu.

Suite à ces évaluations, et au retour positif des utilisateurs ayant manipulé le  $C^3$ , nous avons décidé d'augmenter le nombre d'options disponibles (26 dans cette première version) pour disposer d'un menu complet exploitant le même principe d'interaction. De façon similaire aux menus linéaires 2D, les options de menus ont été hiérarchisées sous la forme d'un graphe. Chaque nœud du graphe utilise le  $C^3$  pour s'afficher, et des mécanismes supplémentaires mais respectueux de l'esprit du paradigme ont été ajoutés au  $C^3$  pour naviguer de menus en sous-menus et inversement. La version hiérarchique du  $C^3$  accepte ainsi un nombre arbitrairement grand d'options, recouvrant l'ensemble des commandes de toute application.

Le  $C^3$  dans sa version simple, remplit le rôle des raccourcis. Le  $C^3$  hiérarchique remplit pour les environnements virtuels celui des menus déroulants de l'interface WIMP, en donnant accès à l'ensemble des fonctions de l'application. Il reste un dernier rôle populaire dans le monde 2D, non transcrit encore par le paradigme du  $C^3$  : les menus contextuels. Ces menus *pop-up* affichent un jeu de fonctions qui dépend d'un contexte précis : la position du pointeur de la souris. Cette idée est aisément étendue au  $C^3$  en affichant son contenu en fonction de la technique de désignation standard sur le Plan de Travail Virtuel, le stylus prolongé d'un rayon virtuel.

Enfin, pour agrémenter le  $C^3$  des dernières aides à la sélection, des infobulles sont ajoutées au menu : des étiquettes flottantes contenant un texte explicatif de l'option

désignée par la position courante du pointeur sphérique.

Le *Command and Control Cube* se présente ainsi comme une solution complète pour environnement virtuel de gestion de fonctions, rapide d'accès, accessible, nécessitant un équipement physique minimum et couvrant les trois grandes techniques de contrôle de menus linéaires : les raccourcis, les menus hiérarchiques et les menus contextuels.

Le **quatrième chapitre** s'intéresse à un manque inhérent aux configurations immersives 3D, qui sont des environnements profondément graphiques et mal adaptés à cette tâche : la saisie de caractères alphanumériques. Le clavier physique ne pouvant être raisonnablement intégré dans la plupart des configurations immersives, nous avons tout d'abord étudié la possibilité d'utiliser un clavier virtuel intégré dans la scène, avec différentes façons de contrôler sa position. Cette première solution ne permet cependant pas une interaction naturelle avec les deux mains, du fait des exigences et des difficultés de précision sur les très nombreuses touches d'un clavier. Une sélection des touches à l'aide de la technique du rayon virtuel a été proposée. Elle reste viable, mais s'apparente à la saisie à un seul doigt, et s'avère nettement plus lente que la saisie traditionnelle.

Nous avons alors constaté une certaine similitude entre les contraintes de notre environnement virtuel pour la saisie de texte et celui des PDA (*Personal Digital Assistant*). Dans les deux cas, l'interaction s'appuie sur un stylo, une zone d'écriture réduite, et un équipement global peu encombrant. Sur les Palm<sup>TM</sup>, une technique d'écriture particulière a suscité notre intérêt : QuikWrite [94] de Ken Perlin. S'appuyant également sur une division de l'espace (ici un plan) en zones, selon des directions simples (gauche, droite, haut, bas), il propose un paradigme d'écriture de caractères minuscules, majuscules et spéciaux (ponctuation, chiffres, etc.) fondé sur la réalisation au stylet de boucles traversant toutes la case centrale. L'écriture par QuikWrite possède également la particularité de s'exécuter sans que le stylet ne quitte la surface d'écriture, ce qui la rapproche de l'écriture manuscrite. Nous avons ainsi développé QuikWrite VR, s'appuyant sur le paradigme de Perlin et l'adaptant aux environnements virtuels. La zone de saisie, très étroite sur un PDA est ici agrandie pour un meilleur confort d'écriture. Notre version profite d'un affichage de la disposition des lettres sur la grille, pour aider l'utilisateur dans son utilisation et apprentissage du système. Lors de l'accès aux jeux étendus de lettres, les caractères majuscules et spéciaux, l'affichage est mis automatiquement à jour pour guider l'utilisateur. Cette zone d'écriture est représentée comme une tablette virtuelle flottant dans l'espace devant

l'utilisateur. Nous avons également choisi de représenter la grille de QuikWrite dans l'espace 3D sous forme de cubes adjacents. Le système d'écriture QuikWrite VR rappelle ainsi l'apparence du Command and Control Cube. Ils partagent également un principe de fonctionnement similaire, le déplacement d'un pointeur à l'intérieur d'une grille 2D ou 3D. QuikWrite VR complète ainsi naturellement le système précédent de contrôle d'application en assurant une homogénéité, profitable à toute interface.

Le stylus du Plan de Travail Virtuel (le stylo équipé d'un capteur) est utilisé pour la manipulation. Une pression sur son bouton marquant le début et la fin de la saisie, et permet de mettre l'écriture "en pause". Les mouvements du stylus contrôlent un avatar de stylo, constraint à rester sur la surface d'écriture, afin de s'affranchir des problèmes de précision. De premiers résultats informels ont montré l'importance d'une phase d'apprentissage pour la disposition des lettres dans QuikWrite VR, mais également une facilité d'utilisation et des progrès rapides.

QuikWrite VR propose ainsi un système simple et rapide à mettre en œuvre de saisie de chaînes de caractères, couvrant l'ensemble des caractères informatiques, pour des environnements virtuels dotés d'un stylus.

Le **cinquième chapitre** présente deux nouvelles techniques d'interaction pour la palette virtuelle de Coquillart et Wesche [31] pour l'exploration de scènes virtuelles 3D, le *Magic Mirror* et les *Magic Windows*, construites sur les métaphores du miroir et de la loupe.

Universellement présents dans notre vie quotidienne, les outils sont des équipements familiers et efficaces pour explorer et interagir avec notre environnement. Parmi les outils destinés à étendre nos capacités de vision, le miroir est un des plus simples et familiers. Le *Magic Mirror* se propose d'importer un miroir dans un univers virtuel comme extension pour explorer l'environnement sans se déplacer. Il ajoute un point de vue supplémentaire dans la scène, manipulable avec simplicité et spontanéité, grâce à la palette tenue en main et la métaphore du miroir.

Le miroir propose une vue inversée qui peut être déconcertante, lorsque l'utilisateur se déplace, où si la vue du miroir est la seule vue disponible. Dans une utilisation conventionnelle, cette vue s'intègre dans le champ de vision d'un utilisateur et l'utilisateur ne se déplace pas pendant qu'il explore la scène à l'aide du miroir. Il bénéficie alors pleinement

de son expérience dans le monde réel avec des miroirs tenus en main.

Un ensemble de propriétés additionnelles, impossibles dans le monde réel, sont adjointes au miroir pour accroître ses possibilités : taille et grossissement variable du miroir, inversion de l'image pour permettre la lecture de textes réfléchis, basculement de la vue de l'utilisateur vers celle du miroir. Une autre fonctionnalité s'attache plus spécifiquement à étendre le champ d'application du miroir en contrôlant son angle de réflexion, à la manière du réglage de l'angle d'ouverture d'un appareil photographique. Le *Magic Mirror* peut capturer la réflexion d'un petit détail jusqu'à un demi-espace entier. Une extension fondamentale, le dégagement de la vue du miroir, s'intéresse à la disponibilité du *Magic Mirror*, comme outil d'exploration. La pyramide de vue, entre l'œil et la surface du miroir est dégagée de tout objet faisant obstruction. Cette fonction assure que le *Magic Mirror* soit utilisable en toutes circonstances. De plus, elle étend considérablement les capacités d'exploration du *Magic Mirror* en permettant d'observer des éléments d'une scène au-delà de surfaces opaques. Le *Magic Mirror* devient ainsi un outil multi-usage capable d'explorer l'intérieur d'objets creux, de voir au-delà des surfaces opaques, de présenter des vues rapides d'un objet sous différents angles sans avoir à se déplacer, et de profiter d'un point de vue supplémentaire à une échelle différente et interactive. Enfin, le *Magic Mirror* intègre dans sa métaphore le principe très vaste des filtres en proposant d'afficher une image réfléchie d'un monde virtuel différent du monde virtuel qui entoure la palette. Il peut par exemple pour l'exploration d'un objet creux comme un tiroir fermé et sombre, proposer une image réfléchie bénéficiant d'un meilleur éclairage, sans altérer la scène principale.

De manière analogue, la métaphore de la loupe et les *Magic Windows* proposent d'utiliser la palette comme support physique pour la localisation de filtres. L'utilisateur regarde à travers la surface transparente de la palette une scène altérée par un filtre. Les *Magic Windows* profitent des extensions précédentes concernant l'ouverture réglable et le dégagement de la vue, pour accroître le pouvoir d'exploration des filtres.

Dans le cadre de cette thèse, et pour soutenir les travaux précédents, nous avons mis en œuvre une plate-forme logicielle complète pour l'affichage et l'interaction sur le Plan de Travail Virtuel. Le **sixième chapitre** présente cette plate-forme intitulée Minify. Elle se compose de deux ensembles indépendants, gérant l'affichage stéréoscopique avec enregistrement des mouvements de la tête et la gestion des outils d'interaction. Le module d'affichage gère le chargement de modèles 3D. Le second module propose un cadre pour

le développement d'outils d'interaction en environnement virtuel. Les équipements et les techniques d'interaction associés sont réunis dans Minifly sous le concept d'outil. Il réunit un ensemble de fonctions ou règles de comportement, et un certain nombre d'attributs optionnels, comme une apparence graphique, une partie physique, un dispositif haptique ou un retour sonore. Minifly régit la création, l'appel, le renvoi et la destruction de ces outils d'interaction. Ils sont attachés au fur et à mesure de l'interaction à deux points d'ancrage logiciel représentant les mains. Seuls les outils attachés à un de ces deux points sont traités à un instant donné.

## 7.2 Travaux futurs et perspectives

Les travaux précédents peuvent être prolongés sur plusieurs voies de recherche. L'étude des performances du menu  $C^3$  a permis d'obtenir une carte 3D des cases les plus accessibles. Pour obtenir une étude plus fine encore, il serait intéressant d'analyser les données de position du pointeur à l'instant de sélection, i.e. quand l'utilisateur relâche le bouton. Les nuages de points correspondant à ces positions dans chaque case peuvent expliquer plus finement les causes des difficultés ou des erreurs de sélection. La morphologie du bras humain joue un rôle important dans la facilité de sélection des différentes cases. Le coude est par exemple un point quasi-fixe dans la plupart des mouvements observables. La question peut se poser de savoir s'il est plus facile pour un utilisateur de réaliser des mouvements par rapport à de grandes directions orthogonales : avant, arrière, gauche, droite, haut, bas, ou selon des sphères concentriques, centrées sur le coude. Il serait dans ce cas envisageable de rémodeler soit la géométrie du  $C^3$ , soit l'application des mouvements de la main sur le pointeur sphérique de sélection.

Suite aux premières études concernant les performances de sujets ayant un premier contact avec le  $C^3$ , les analyses futures doivent porter sur la courbe d'apprentissage du  $C^3$ , qui comprend entre autre le temps d'apprentissage et la recherche d'un palier de performances.

L'étude du  $C^3$  a montré la validité de son approche en tant que menu à accès rapide pour environnement virtuel immersif. Il serait intéressant pour mieux situer le  $C^3$  de comparer ses performances à d'autres menus du même type ou à ses équivalents du monde 2D, selon le mode de sélection, normal ou hiérarchique, et selon les différentes données

manipulées : quelconques, sans ordre précis comme une liste de couleurs, classées linéairement comme par exemple une liste de chiffres, ou ordonnées selon deux dimensions, ou trois dimensions voire plus encore.

Le  $C^3$  a été développé sur un Plan de Travail Virtuel. Une réflexion doit être menée pour l'étendre à des configurations de réalité virtuelle ne possédant pas d'axes privilégiés visibles pour l'alignement des cubes. Il faut étudier les différentes approches possibles, pour choisir la plus pertinente : alignement sur les épaules de l'utilisateur, sur la direction de son avant-bras, ou sur la direction de regard.

La technique d'écriture QuikWrite VR présentée dans ce mémoire, évaluée au cours de tests informels, nécessite des analyses plus approfondies pour déterminer les meilleurs emplacements des lettres selon les langues (notamment la langue française), évaluer la courbe d'apprentissage du menu et mesurer les performances moyennes des utilisateurs.

Comme suggéré par un des sujets, le plan de QuikWrite VR doit être basculé horizontalement pour diminuer la fatigue du bras qui écrit. Ce bras est généralement non soutenu dans un environnement virtuel immersif. Il peut être intéressant d'évaluer l'apport d'un support physique à cet égard, au niveau du plan d'écriture. Un support physique peut se présenter sous la forme d'une plaque transparente tenue en main comme une palette, ou tenue par un bras fixe, comme un pupitre. Une solution alternative est à l'étude, s'appuyant sur un dispositif à retour d'effort à base de fils tendus, accrochés à la pointe du stylo. Un système de contrôle permet de régler la tension des fils pour simuler auprès de la main tenant le stylo l'illusion de forces. Dans notre cas, le Spidar permet de simuler un plan à l'emplacement de la surface plane graphique de QuikWrite VR.

De même que pour le menu  $C^3$ , une comparaison du système de saisie de texte QuikWrite VR avec les quelques solutions concurrentes serait nécessaire pour situer la technique et ses avantages.

QuikWrite VR et le  $C^3$  forment deux outils homogènes pour le contrôle d'application en environnement virtuel. D'autres éléments d'interface peuvent venir compléter les domaines du contrôle d'application déjà couverts par ces deux solutions, ou proposer des réponses plus spécifiques à certaines tâches de contrôle. Il est naturel de regarder les éléments d'interface existant dans les paradigmes de contrôle d'application 2D, et de chercher à proposer des solutions parallèles pour les environnements virtuels. Cette dé-

marque permet de s'assurer que les interfaces immersives peuvent bénéficier au minimum des mêmes moyens de contrôle que celles des stations de travail, et de faciliter le portage des applications 2D vers les environnements virtuels.

Il pourrait par exemple être intéressant de rechercher des solutions pour fixer un ou plusieurs paramètres continus au sein d'un environnement virtuel. La technique des règlettes ou *sliders* en anglais couvre cet aspect dans les interfaces 2D. La présence d'une dimension supplémentaire dans les environnements immersifs peut ici être un atout pour le contrôle simultané de plusieurs paramètres continus corrélés. Il existe aussi de nombreuses tâches de sélection 1D revêtant des aspects plus spécialisés que le menu linéaire traditionnel. Beaucoup d'applications fonctionnent sur le principe de modes et de transitions d'état à un autre. Pour visualiser l'état courant, activer ou désactiver un mode, des composants d'interface de type interrupteur simple (case à cocher) ou à plusieurs états (*radio buttons*) existent notamment. Bien que le  $C^3$  permette de réaliser cette tâche de sélection discrète, il ne permet pas actuellement d'afficher l'état courant. Une mise en œuvre plus spécifique peut-être préférable. De même des opérations de sélection dans une liste linéaire de grande taille, comme par exemple la recherche d'un fichier dans un répertoire, est généralement réalisée par un défilement avec la technique de l'ascenseur. Elle peut être simulée sur le  $C^3$  en réservant une des 26 cases au renouvellement des 25 autres. La encore, une solution spécifique peut être plus confortable.

A l'opposé des méthodes de communication de l'utilisateur vers l'application, il faut également étudier les méthodes de communication de l'application vers l'utilisateur. En 2D, les boîtes de dialogue, les icônes d'état, les barres d'état sont des méthodes appropriées du fait de la métaphore plane du bureau. Il est possible de s'interroger sur la meilleure façon de réaliser cette communication dans un environnement 3D. Introduire un texte 2D dans l'univers virtuel, amène les mêmes considérations de placement que celles vues dans l'état de l'art pour les menus linéaires 2D. D'autres modalités, comme la voix peuvent également être envisagées.

A plus long terme, les interfaces de contrôle d'application en environnement virtuel se dirigeront probablement vers des solutions multi-modales, exploitant les divers sens rendus disponibles par l'immersion. Une constatation naïve sur les utilisateurs immergés est qu'ils cherchent spontanément à agir avec les mains. La ressemblance avec le monde réel et la possibilité d'agir directement incitent à employer des gestes courants, simples, pour

contrôler son environnement. Pour le contrôle d'application des gestes mesurés, simples et de faible amplitude, seront sans doute privilégiés, pour éviter la fatigue de la répétition. En effet, les configurations de réalité virtuelle demandent souvent aux utilisateurs d'être debout et contrairement aux stations de travail proposant des périphériques (clavier et souris) avec une surface de repos pour les bras, en environnement immersif l'interaction est plus sollicitante. Les interfaces graphiques resteront probablement également un pilier incontournable du contrôle d'application. Ce type d'interface offre le meilleur retour d'information, par le sens le plus évolué de l'humain, la vue. Pour l'apprentissage, pour l'aide, pour la communication générale d'informations de l'application vers l'homme, l'interface graphique reste la plus souple et la plus puissante. En complément de ces deux modalités principales, l'ajout de la voix est une solution intéressante lorsqu'elle vient en soutien des faiblesses de l'une ou l'autre des techniques gestuelle et graphique. L'utilisation intensive de la voix reste fatiguante, et contraignante pour la communication avec d'autres utilisateurs.

Les techniques de sélection/manipulation et de navigation sont très souvent liées à des tâches précises et spécifiques de l'application. En conséquence une diversité des techniques d'interaction risque de s'imposer face à la recherche d'une solution universelle générique. Les techniques les plus performantes présenteront des caractéristiques optimisées pour un contexte donné, un type de travail voire un corps de métier. Le contrôle d'application au contraire est indépendant de la tâche à réaliser, et s'inscrit dans une logique différente. Il doit être centré sur l'utilisateur, qui doit trouver quel que soit le système sur lequel il travaille un environnement familier, transparent et maîtrisé. En réalisant cet objectif, l'interface immersive générique découpe les outils propres à l'application, des moyens d'en disposer. L'utilisateur peut alors découvrir chaque nouvelle application et ses outils dans un cadre connu. Le *C<sup>3</sup>* a ainsi été guidé tout au long de sa conception par un souci de proposer un paradigme de contrôle simple, rapide et universel.

## Glossaire

**Avatar** Un élément graphique identifiant et représentant dans un environnement immersif une entité active du monde réel, par exemple une personne, un bras ou un objet donné tenu en main.

**Configuration de réalité virtuelle** Une installation physique permettant la stimulation des sens d'un ou plusieurs utilisateurs, pour la création d'un environnement virtuel.

**Environnement virtuel** C'est un univers fictif à trois dimensions, perçu par un ou plusieurs utilisateurs comme les entourant totalement ou partiellement, avec lequel ils peuvent interagir.

**Frustum** Terme latin signifiant morceau ou fragment. Le frustum désigne une portion du champ de vision, pyramidale, utilisée dans les calculs informatiques de vues.

**Kinesthésique** Relatif à la kinesthésie ; qui concerne la sensation de mouvement des parties du corps.

**Menu pop-up** (terme anglais) Menu graphique apparaissant sur commande. Il est invisible à l'état de repos.

**Métaphore** Dans le contexte de l'interaction en environnement virtuel, la métaphore désigne une entité conceptuelle servant à l'interaction, au moyen d'un terme, qui en signifie une autre, en vertu d'une analogie entre les deux entités.

**Palette** Périphérique d'interaction en réalité virtuelle. Elle se présente sous la forme d'une raquette composée d'une surface transparente rectangulaire munie d'un manche et d'un capteur de position et d'orientation.

**Paradigme** Dans le contexte de l'interaction en environnement virtuel, le paradigme désigne le concept qui sous-tend un processus d'interaction, composé d'une ou plusieurs étapes.

**Plan de Travail Virtuel** Configuration de réalité virtuelle semi-immersive. Le Plan de Travail Virtuel ou *Workbench* en anglais est une table sur la surface de laquelle est affichée une image stéréoscopique. Certaines configurations de Plan de Travail Virtuel proposent un écran supplémentaire contigu, vertical, pour augmenter le champ de vision.

**Prop** (terme anglais) Outil utilisé en réalité virtuelle comme médium pour l'interaction.

**Proprioception** Perception qu'a l'homme de son propre corps, par les sensations kinesthésiques et posturales en relation avec la situation du corps par rapport à l'intensité de l'attraction terrestre.

**Stylus** Périphérique d'interaction en réalité virtuelle. Il se présente sous la forme d'un stylo équipé d'un capteur de position et d'orientation, et d'un ou plusieurs boutons sur sa longueur.

**Widget** (terme anglais) Elément d'interface associant un comportement à un composant graphique.



# Bibliographie

- [1] I. Angus and H. Sowizral. Embedding the 2d interaction metaphor in a real 3d virtual environment. In *Proceedings of SPIE, Stereoscopic Displays and Virtual Reality Systems*, volume 2409, pages 282–293, 1995.
- [2] Ravin Balakrishnan and Ken Hinckley. The role of kinesthetic reference frames in two-handed input performance. In *Proceedings of the ACM Symposium on User Interface Software and Technology, Two Hands*, pages 171–178, 1999.
- [3] Ravin Balakrishnan and Ken Hinckley. Symmetric bimanual interaction. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*, volume 1 of *Multi-Hand + Multi-DOF*, pages 33–40, 2000. 869 KB.
- [4] T. Baudel and M. Beaudouin-Lafon. Charade : Remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7) :28–35, July 1993.
- [5] Michel Beaudouin-Lafon. Instrumental interaction : an interaction model for designing post-WIMP user interfaces. In *CHI*, pages 446–453, 2000.
- [6] E.A. Bier, M.C. Stone, K. Pier, W. Buxton, and T.D. DeRose. Toolglass and magic lenses : The see-through interface. In *SIGGRAPH '93, Computer Graphics*, pages 73–80, August 1993.
- [7] Eric Bier, Maureen Stone, Ken Fishkin, William Buxton, and Thomas Baudel. A Taxonomy of See-Through Tools. In *BIBCHI*, pages 358–364. Addison-Wesley, April 1994.
- [8] Chuck Blanchard, Scott Burgess, Young Harvill, Jaron Lanier, Ann Lasko, Mark Oberman, and Michael Teitel. Reality built for two : A virtual reality tool. In Rich Riesenfeld and Carlo Sequin, editors, *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, volume 24, pages 35–36, March 1990.

- [9] G. Blaskó and S. Feiner. A menu interface for wearable computing. In *6th International Symposium on Wearable Computers ISWC'2002*, pages 164–165, 2002.
- [10] C. H. Blinkenstorfer. Graffiti. *Pen Computing*, January 1995.
- [11] K. Boff, L. Kaufman, and J. Thomas. Handbook of perception and human performance, 1986.
- [12] R. A. Bolt and E. Herranz. Two-handed gesture in multi-modal natural dialog. In *Proc. ACM SIGGRAPH Symposium on User Interface Software and Technology (UIST'92)*, pages 7–13, 1992.
- [13] Richard A. Bolt. Put-that-there : Voice and gesture at the graphics interface. *Computer Graphics*, pages 262–270, August 1980.
- [14] M. Bordegoni and M. Hemmje. A dynamic gesture language and graphical feedback for interaction in a 3D user interface. *Computer Graphics Forum*, 12(3) :C1–C11, 1993.
- [15] J. Boritz and K. S. Booth. A study of interactive 3d point location in a computer simulated virtual environment. In *Proceedings of VRST'97*, pages 191–187, 1997.
- [16] D. Bowman, V. Ly, and C.J. Campbell. Pinch keyboard : Natural text input for immersive virtual environments. Technical Report TR-01-15, Virginia Tech Dept. of Computer Science, 2001.
- [17] D. A. Bowman, J. LaViola, M. Mine, and I. Poupyrev. Advanced topics in 3d user interface design. In *Course Notes - SIGGRAPH 2001*, 2001.
- [18] D.A. Bowman and C.A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Proc. IEEE VR'2001*, 2001.
- [19] Doug Bowman, Donald Johnson, and Larry F. Hodges. Testbed evaluation of VE interaction techniques. In Mel Slater, editor, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST-99)*, pages 26–33, N.Y., December 20–22 2000. ACM Press.
- [20] Doug A. Bowman and Larry F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Symposium on Interactive 3D Graphics*, pages 35–38, 182, 1997.
- [21] D. C. Brogan, R. A. Metoyer, and J. K. Hodgins. Dynamically simulated characters in virtual environments. *IEEE Computer Graphics and Applications*, 15(5) :58–69, September/October 1998.

- [22] F. P. Brooks. Grasping reality through illusion : Interactive graphics serving science. In *Proc. ACM CHI'88 Human Factors in Computing Systems Conference*, pages 1–11, 1988.
- [23] J. Butterworth, A. Davidson, S. Hench, and T. M. Olano. 3dm : A three dimensional modeler using a head-mounted display. In *Proc. 1992 Symposium on Interactive 3D Graphics*, pages 135–138, 1992.
- [24] D. Jessome C. Ware. Using the bat : a six-dimensional mouse for object placement. *8(6) :65–70*, 1988.
- [25] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems, Menus*, pages 95–100, 1988.
- [26] CAVE5D. <http://www.ccpo.odu.edu/~cave5d/homepage.html>.
- [27] O. Chenu. Rapport de dess "portage de quikwrite vr sur open inventor immersif", i3d inria rhones alpes, 2003.
- [28] Paolo Cignoni, Claudio Montani, and Roberto Scopigno. Magicsphere : an insight tool for 3d data visualization. *Computer Graphics Forum*, 13(3) :317–328, 1994.
- [29] P. R. Cohen. The role of natural language in a multimodal interface. In *ACM UIST'92 Symp. on User Interface Software & Technology*, pages 143–149. 1992.
- [30] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. *Computer Graphics*, 25(2) :183–188, March 1992.
- [31] S. Coquillart and G. Wesche. The virtual palette and the virtual remote control panel : A device and an interaction paradigm for projection-based virtual environments. In *IEEE VR'99*, 1999.
- [32] C. Cruz-Neira, D. Sandin, and T. DeFanti. Surround-screen projection-based virtual reality : The design and implementation of the CAVE. In *Computer Graphics (SIGGRAPH Proceedings)*, pages 135–142, August 1993.
- [33] L. Cutler, B. Fröhlich, and P. Hanrahan. Two-handed direct manipulation on the responsive workbench. In *1997 Symposium on Interactive 3D Graphics*, 1997.
- [34] R. de Amicis, M. Fiorentino, and A. Stork. Parametric interaction for cad application in virtual reality environment. In *ADM Associazione Nazionale Disegno di Macchine : International Conference on Design Tools and Methods in Industrial Engineering*, pages D3/43–D3/52, 2001.

- [35] Michael F. Deering. HoloSketch : a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction*, 2(3) :220–238, September 1995.
- [36] Virtual Director. <http://virdir.ncsa.uiuc.edu/virdir/>.
- [37] K. Dorfmüller-Ulhaas and D. Schmalstieg. Finger tracking for interaction in augmented environments, 2001.
- [38] D. Bowman et L. Hodges. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *The Journal of Visual Languages and Computing*, 10(1) :37–53, 1999.
- [39] F. Evans, S. Skiena, and A. Varshney. Vtype : Entering text in a virtual world, 1999. submitted to International Journal of Human-Computer Studies.
- [40] Fakespace. <http://www.fakespacelabs.com/>.
- [41] Inc.) (Fakespace. Pinch gloves.
- [42] Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. Windows on the world : 2D windows for 3D augmented reality. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Virtual Reality, pages 145–155, 1993.
- [43] J. D. Foley, V. Wallace, and P. Chan. The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, pages 13–48, November 1984.
- [44] Andrew S. Forsberg, Kenneth P. Herndon, and Robert P. Zeleznik. Aperture based selection for immersive virtual environments. In *ACM Symposium on User Interface Software and Technology*, pages 95–96, 1996.
- [45] A.S. Forsberg, J.J. LaViola, and R.C. Zeleznik. Ergodesk : A framework for two- and three-dimensional interaction at the activedesk. In *Second International Immersive Projection Technology Workshop*, Ames, Iowa, May 1998.
- [46] Bernd Fröhlich and John Plate. The cubic mouse : a new device for three-dimensional input. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 526–531. ACM Press, 2000.
- [47] P. Fuchs, G. Moreau, and J.-P. Papin. *Le Traité de la Réalité Virtuelle*. Les Presses de L’Ecole des Mines de Paris, 2001.
- [48] P. Fuchs, G. Moreau, and J.-P. Papin. *Le Traité de la Réalité Virtuelle - Seconde Edition*. Les Presses de L’Ecole des Mines de Paris, 2003.

- [49] Massaki Fukumoto and Yoshinobu Tonomura. "body coupled fingering" : Wireless wearable keyboard. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS : Input & Output in the Future*, pages 147–154, 1997.
- [50] A. Goldberg and D. Robson. A metaphor for user interface design. In *Proceedings of the University of Hawaii Twelfth Annual Symposium on System Sciences*, pages 148–157., 1979.
- [51] Scott B. Grissom and Gary Perlman. StEP(3D) : A standardized evaluation plan for three-dimensional interaction techniques. *International Journal of Human-Computer Studies*, 43(1) :15–41, 1995.
- [52] J. Grosjean, J.-M. Burkhardt, S. Coquillart, and P. Richard. Evaluation of the command and control cube. In *International Conference on Multimodal Interfaces (ICMI'2002)*, Pittsburgh, USA, October 2002.
- [53] J. Grosjean and S. Coquillart. The magic mirror : A metaphor for assisting the exploration of virtual worlds. In *Spring Conference on Computer Graphics SCCG'1999*, Bratislava, Slovakia, April 1999.
- [54] J. Grosjean and S. Coquillart. Command & control cube : a shortcut paradigm for virtual environments. In B. Fröhlich, J. Deisinger, and H.-J. Bullinger, editors, *Proceedings of the joint 5th Immersive Projection Technology and 7th Eurographics Virtual Environments Workshop (EGVE-01)*, pages 1–12, Wien, Austria, May 16–18 2001. Springer-Verlag.
- [55] J. Grosjean and S. Coquillart. Contrôle d'application en environnement virtuel : le command and control cube. In *XVèmes journées de l'association française d'informatic graphique (AFIG'2002)*, Lyon, France, Décembre 2002.
- [56] J. Grosjean and S. Coquillart. Quikwriting on the responsive workbench. In *Technical Sketch SIGGRAPH'2002*, San Antonio, USA, July 2002.
- [57] Y. Guiard. Asymmetric division of labor in human skilled bimanual action : The kinematic chain as a model. *The Journal of Motor Behavior*, 19(4) :486–517, 1987.
- [58] M. Hachet and P.Guitton. The interaction table - a new input device designed for interaction in immersive large display environments. *Proceedings of the Eighth Eurographics Workshop on Virtual Environments*, pages 189–196, May 2002.
- [59] C. Hand. A survey of 3d interaction techniques. *Computer Graphics Forum*, 16(5) :269–281, December 1997.

- [60] A. G. Hauptmann. Speech and gestures for graphic image manipulation. In *Proc. ACM CHI'89 Conference on Human Factors in Computing Systems*, pages 241–245, 1989.
- [61] K. Herndon, A. van Dam, and M. Gleicher. The challenges of 3d interaction : A CHI'94 workshop. *SIGCHI Bulletin*, 26(4) :36–43, October 1994.
- [62] K. Hinckley, R. Pausch, J.C. Goble, and N.F. Kassell. Passive real-world interface props for neurosurgical visualization. In *CHI'94*, volume 1 of *Interacting in 3-D*, pages 452–458, 1994.
- [63] Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. Usability analysis of 3d rotation techniques. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 1–10. ACM Press, 1997.
- [64] J.M. Hollerbach. Locomotion interfaces. *Handbook of Virtual Environments Technology*, pages 239–254, 2002.
- [65] J.M. Hollerbach, Y. Xu, R. Christensen, and S.C. Jacobsen. Design specifications for the second generation sarcos treadport locomotion interface. In *Proc. ASME Dynamic Systems and Control Division, DSC-Vol. 69-2*, pages 1293–1298, November 2000.
- [66] MultiGen Inc. Smartscene(tm).
- [67] P. Kabbash, I. S. MacKenzie, and W. Buxton. Human performance using computer input devices in the preferred and non-preferred hands. In *INTERCHI'93 Conference on Human Factors in Computing Systems*, pages 474–481, 1993.
- [68] D. B. Koons, C. J. Sparrell, and K. R. Thorisson. Integrating simultaneous output from speech, gaze, and hand gestures. In *M. Maybury, (Ed.). Intelligent Multimedia Interfaces, Menlo Park : AAAI/MIT Press*, pages 243–261, 1993.
- [69] David B. Koons and Carlton J. Sparrell. ICONIC : Speech and depictive gestures at the human-machine interface. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 2 of *VIDEOS : Part II – Techniques for Improved Human-Computer Interaction*, pages 453–454, 1994.
- [70] G. E. Krasner and S. T. Pope. A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80. *Journal of Object Oriented Programming*, 1(3) :26–49, August/September 1988.

- [71] W. Krüger, C.A. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The responsive workbench : A virtual work environment. *IEEE Computer*, pages 42–48, July 1995.
- [72] W. Krüger and B. Fröhlich. The responsive workbench. *IEEE Computer Graphics and Applications*, pages 12–15, May 1994.
- [73] W. Kuhn and B. Blumenthal. Spatialization : Spatial metaphors for user interfaces. In *Course Notes - CHI'96*, 1996.
- [74] Gordon Kurtenbach and William Buxton. User learning and performance with marking menus. In Beth Adelson, Susan Dumais, and Judith Olson, editors, *Proceedings of the Conference on Human Factors in Computing Systems*, pages 258–264, New York, NY, USA, April 1994. ACM Press.
- [75] Gordon P. Kurtenbach, Abigail J. Sellen, and William A. S. Buxton. An empirical evaluation of some articulatory and cognitive aspects of marking menus. *Human-Computer Interaction*, 8(1) :1–23, 1993.
- [76] J. Laviola. Msvt : A virtual reality-based multimodal scientific visualization tool, 1999.
- [77] Anatole Lecuyer. Contribution à l'étude des retours haptique et pseudo-haptique et de leur impact sur les simulations d'opérations de montage/démontage en aéronautique. Master's thesis, Université Paris XI, Paris.
- [78] J. Liang, M. Green, 3rd International Conference on CAD, and Computer Graphics. JDCAD : A highly interactive 3d modeling system, August 1993.
- [79] J. Liang, C. Shaw, and M. Green. On temporal-spatial realism in the virtual reality environment. In *Proc. ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 19–25, 1991. (UIST '91).
- [80] Jiandong Liang and Mark Green. JDCAD : A highly interactive 3D modeling system. *Computers and Graphics*, 18(4) :499–506, July–August 1994.
- [81] R. Lindeman, J. Sibert, and J. Hahn. Hand-held windows : Towards effective 2d interaction in immersive virtual environments. In *IEEE VR'99*, 1999.
- [82] R. Lindeman, J. Sibert, and J. Templeman. The effect of 3d widget representation and simulated surface constraints on interaction in virtual environments. In *IEEE VR'2001*, pages 141–148, 2001.

- [83] I MacKenzie, . S., and W. Buxton. Extending fitts' law to two-dimensional tasks. In *Proc. CHI '92 Conference on Human Factors in Computing Systems*, pages 219–226, 1992.
- [84] J. Mackinlay, S. Card, and G. Robertson. Rapid controlled movement through a virtual 3d workspace. *Computer Graphics*, 24(4) :171–176, 1990.
- [85] D. Mapes and J. Moshell. A two-handed interface for object manipulation in virtual environments. *Presence : Teleoperators and Virtual Environments*, 4(4) :403–416, 1995.
- [86] Michael Meehan, Brent Insko, Mary Whitton, and Jr. Frederick P. Brooks. Physiological measures of presence in stressful virtual environments. In John Hughes, editor, *SIGGRAPH 2002 Conference Proceedings*, Annual Conference Series, pages 645–652. ACM Press/ACM SIGGRAPH, 2002.
- [87] M. Mine. Virtual environment interaction techniques. *SIGGRAPH'95 Course*, (8), 1995.
- [88] Mark R. Mine, Frederick P. Brooks, Jr., and Carlo H. Séquin. Moving objects in space : Exploiting proprioception in virtual-environment interaction. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 19–26. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.
- [89] Ring Mouse. <http://www.worklink.net/ringmouse.html>.
- [90] R. O'Hagan and A. Zelinsky. Finger track - a robust and real-time gesture interface, 1997.
- [91] R. O'Hagan, A. Zelinsky, and S. Rougeaux. Visual gesture interfaces to virtual environments. *Interacting with Computers, Special Issue*, 2002.
- [92] L. Arnal. P. Bourdot, M. Dromigny. Virtual navigation fully controlled by head tracking. *Actes des Rencontres Réalité Virtuelle de Laval*, 1999.
- [93] A. Paljic, J.-M. Burkhardt, and S. Coquillart. A study of distance of manipulation on the responsive workbench. In *IPT'2002 Symposium, Orlando, US*, pages 251–258, October 2002.
- [94] Ken Perlin. Quikwriting : Continuous stylus-based text entry. In *Proceedings of the ACM Symposium on User Interface Software and Technology, Fast Pen Input*, pages 215–216, 1998.

- [95] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 39–ff. ACM Press, 1997.
- [96] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls : seamless interaction at multiple scales in virtual environments. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 141–145. ACM Press, 1999.
- [97] I. Poupyrev, N. Tomokazu, and S. Weghorst. Virtual notepad : Handwriting in immersive vr, 1998.
- [98] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa. A framework and testbed for studying manipulation techniques for immersive VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST-97)*, pages 21–28, New York, September 15–17 1997. ACM Press.
- [99] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique : Non-linear mapping for direct manipulation in VR. In *Proceedings of the ACM Symposium on User Interface Software and Technology, Papers : Virtual Reality (TechNote)*, pages 79–80, 1996.
- [100] Ivan Poupyrev, Suzanne Weghorst, and Sidney Fels. Non-isomorphic 3d rotational techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 540–547. ACM Press, 2000.
- [101] McMillan G. Eggelston R. and Anderson T. Nonconventional controls. In *Handbook of human factors and ergonomics*, pages 729–771, 1997.
- [102] Stephane Redon. Master's thesis, 2002.
- [103] Sarcos. Locomotion interfaces.
- [104] Dieter Schmalstieg, L. Miguel Encarnaçāo, and Zsolt Szalavári. Using transparent props for interaction with the virtual table. In *Symposium on Interactive 3D Graphics*, pages 147–153, 1999.
- [105] Chris M. Schmandt. Spatial input/display correspondence in a stereoscopic computer graphic work station. In *Proc. ACM SIGGRAPH '83*, pages 253–262, 1983. Published as Computer Graphics, volume 17, number 3.
- [106] A. Seay, D. Krum, W. Ribarsky, and L. Hodges. Multimodal interaction techniques for the virtual workbench.

- [107] Chris Shaw and Mark Green. THRED : A two-handed design system. *Multimedia Systems*, 5(2) :126–139, 1997.
- [108] M. Slater and M. Usoh. An experimental exploration of presence in virtual environments. Technical Report QMW-DCS-1993-689, Queen Mary and Westfield College, Department of Computer Science, December 1993.
- [109] M. Slater, M. Usoh, and Y. Chrysanthou. The influence of dynamic shadows on presence in immersive virtual environments. In M. Göbel, editor, *Virtual Environments '95*, Eurographics, pages 8–21. Springer-Verlag Wien New York, 1995.
- [110] M. Slater, M., Usoh and A. Steed. Depth of presence in virtual environments. *Presence : Teleoperators and Virtual Environments*, 3.2 MIT Press :130–144, 1994.
- [111] Timo Sowa and Ipke Wachsmuth. Interpretation of shape-related iconic gestures in virtual environments. *Lecture Notes in Computer Science*, 2298 :21–??, 2002.
- [112] T. Stein and S. Coquillart. The metric cursor. In Brian A. Barsky, Yoshihisa Shinagawa, and Wenping Wang, editors, *Proceedings of the 8th Pacific Graphics Conference on Computer Graphics and Application (PACIFIC GRAPHICS-00)*, pages 381–386, Los Alamitos, CA, October 3–5 2000. IEEE.
- [113] R. Stoakley, M. Conway, and R. Pausch. Virtual reality on a WIM : Interactive worlds in miniature. In *CHI'95*, pages 265–272. 1995.
- [114] Stanislav L. Stoev and Dieter Schmalstieg. Application and taxonomy of through-the-lens techniques. In Hanqiu Sun and Qunsheng Peng, editors, *Proceedings of the 2002 ACM Symposium on Virtual Reality Software and Technology (VRST-02)*, pages 57–64, New York, November 11–13 2002. ACM Press.
- [115] D. Sturman, D. Zeltzer, and S. Pieper. Hands-on interaction with virtual environments. In *Proc. ACM Symposium on User Interface Software and Technology*, pages 19–24., 1989.
- [116] Martin Usoh, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, Frederick P. Brooks, and Jr. Walking > walking-in-place > flying, in virtual environments. *SIGGRAPH'99*, 1999.
- [117] Ron van Teylingen, William Ribarsky, and Charles van der Mast. Virtual data visualizer. *IEEE Transactions on Visualization and Computer Graphics*, 3(1) :65–74, January – March 1997. ISSN 1077-2626.
- [118] J. Viega, M.J. Conway, G. Williams, and R. Pausch. 3d magic lenses. In *UIST'96*, 1996.

- [119] Colin Ware and Ravin Balakrishnan. Reaching for objects in VR displays : lag and frame rate. *ACM Transactions on Computer-Human Interaction*, 1(4) :331–356, 1994.
- [120] K. Watsen, R. Darken, and M. Capps. A handheld computer as an interactiondevice to a virtual environment. In *Proceedings of Third Immersive Projection Technology Workshop*, 1999.
- [121] D. Weimer and S. K. Ganapathy. A synthetic visual environment with hand gesturing and voice input. In *CHI'89 Conference on Human Factors in Computing Systems*, pages 235–240, 1989.
- [122] G. Wesche and M. Droske. Conceptual free-form styling on the responsive work-bench. In *VRST 2000*, 2000.
- [123] S. A. Weyer and A. H. Borning. A prototype electronic encyclopedia. *ACM Transactions on Office Information Systems*, 3(1) :63–88, January 1985.
- [124] Matthias M. Wloka and Eliot Greenfield. The virtual tricorder : A uniform interface for virtual reality. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Virtual and Augmented Realities, pages 39–40, 1995. TechNote.
- [125] R. Zeleznik, A. Forsberg, and P. Strauss. Two pointer input for 3D interaction. pages 115–120, 1997.
- [126] Robert Zeleznik, Timothy Miller, and Andrew Forsberg. Pop through mouse button interactions. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 195–196. ACM Press, 2001.
- [127] Robert C. Zeleznik, Kenneth P. Herndon, Daniel C. Robbins, Nate Huang, Tom Meyer, Noah Parker, and John F. Hughes. An interactive 3D toolkit for constructing 3D widgets. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 81–84, 1993.
- [128] S. Zhai, P. Milgram, and W. Buxton. The effects of using fine muscle groups in multiple degree-of-freedom input. In *Proc. ACM CHI'96 Conference on Human Factors in Computing Systems*, pages 308–315, 1996.
- [129] Shumin Zhai, William Buxton, and Paul Milgram. The "silk cursor" : Investigating transparency for 3d target acquisition. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, volume 1 of *Interacting in 3-D*, pages 459–464, 1994.

- [130] T. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. A hand gesture interface device. In *Proc. of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, pages 189–192, 1987.