

Etude de l'infuence de la séparation des degrés de liberté pour la manipulation 3-D à l'aide de surfaces tactiles multipoints

Anthony Martinet

▶ **To cite this version:**

Anthony Martinet. Etude de l'infuence de la séparation des degrés de liberté pour la manipulation 3-D à l'aide de surfaces tactiles multipoints. Interface homme-machine [cs.HC]. Université des Sciences et Technologie de Lille - Lille I, 2011. Français. NNT : . tel-00638449v2

HAL Id: tel-00638449

https://theses.hal.science/tel-00638449v2

Submitted on 18 Jan 2013

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

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

Université Lille1
Sciences et Technologies

École doctorale Sciences Pour l'Ingénieur
Numéro d'ordre : 40584

Étude de l'influence de la séparation des degrés de liberté pour la manipulation 3-D à l'aide de surfaces tactiles multipoints

THÈSE

présentée et soutenue publiquement le 3 octobre 2011

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille

(spécialité informatique)

par

Anthony Martinet

Composition du jury

Président :	Pr. Jean-Christophe Routier
Rapporteurs :	Pr. Dominique Bechmann Dr. Anatole Lecuyer
Examinateurs :	Dr. Martin Hachet Dr. François Bérard
Directeurs de thèse :	Pr. Laurent Grisoni Dr. Géry Casiez

Laboratoire d'Informatique Fondamentale de Lille — UMR CNRS 8022

lifl Laboratoire d'Informatique Fondamentale de Lille

Mis en page avec la classe thloria.

Résumé

Inventées au milieu des années 1980, nous assistons actuellement au développement à grande échelle des interfaces tactiles multipoints. Par comparaison avec la souris, les écrans multipoints nécessitent un nouveau processus d'étude afin de comprendre les changements apportés par cette technologie. Le travail déjà effectué avec d'autres périphériques d'entrée doit être repris à zéro afin de formuler de nouvelles théories pour la création d'interfaces tactiles multipoints.

La manipulation à trois dimensions (3-D) d'un objet nécessite le contrôle de six degrés de liberté. Trois sont responsables de la position de l'objet et trois contrôlent son orientation. Dans ce mémoire, nous nous focalisons sur la manipulation 3-D d'un objet à l'aide de surfaces tactiles multipoints. Plus précisément, nous avons étudié le lien entre le périphérique d'entrée (ici la surface multipoint) et la tâche à effectuer (ici la manipulation 3-D).

En premier lieu, nous avons introduit une taxonomie pour la représentation des techniques d'interaction 3-D spécifiques aux écrans multipoints. Cette taxonomie nous a permis de mettre en avant l'influence des stratégies de conception des techniques en termes d'intégration de degrés de liberté.

Par la suite, nous avons mené une étude concernant les techniques d'interactions permettant le positionnement 3-D d'un objet. Nous avons présenté deux nouvelles techniques, tirant profit des capacités offertes par les écrans multipoints. L'évaluation de ces techniques nous révéla des conclusions quant à la coordination de l'utilisateur dans la réalisation de la tâche.

Enfin, nous avons introduit une nouvelle technique d'interaction de manipulation 3-D. Cette technique a été évaluée plus rapide que les techniques existantes et nous avons pu déterminer les raisons expliquant cette performance accrue.

Ces travaux, qui s'intéressent à la coordination de l'utilisateur lors de la réalisation de tâches 3-D à l'aide de surfaces multipoints, recommandent une approche favorisant la séparation des degrés de liberté, dans le but de pouvoir paralléliser les actions de l'utilisateur afin de tirer profit de la bande passante d'entrée plus importante offerte par les écrans multipoints.

Mots-clés: manipulation 3-D, technique d'interaction, écran multipoint, intégration et séparation de degrés de liberté, évaluation.

Abstract

Invented in the mid-1980s, we are currently witnessing large-scale development of multitouch interfaces. Compared with the traditional mouse, multitouch displays require rethinking from scratch interaction techniques in order to understand the changes brought by this technology and formulate new theories.

Manipulating three-dimensional (3-D) objects require the control of six degrees of freedom. Three are responsible for the object's position and three control the orientation. In this work, we focused on 3-D manipulation of an object using multitouch displays. Specifically, we studied the relationship between the input device (the multitouch display) and the task (3-D manipulation).

First, we introduced a taxonomy for the representation of 3-D interaction techniques specific to multitouch displays. This taxonomy allowed us to highlight the influence of design strategies in terms of integration of degrees of freedom.

Subsequently, we conducted a study of interaction techniques for 3-D positioning of an object. We presented two new techniques, which take advantage of capabilities offered by multitouch displays. The evaluation of these techniques revealed us results regarding the user coordination in realizing the task.

Finally, we introduced a new interaction technique to handle 3-D manipulation. This technique was evaluated faster than existing techniques and we were able to determine the reasons for the improved performance.

This work, which focus on the user coordination when using multitouch displays to perform 3-D tasks, recommend an approach that favors the separation of degrees of freedom, in order to parallelize the user actions so as to take advantage of the extra input bandwidth offered by this type of input devices.

Keywords: 3D manipulation, interaction technique, multitouch screen, integration and separation of degrees of freedom, evaluation.

Remerciements

En tout premier lieu, je souhaite remercier Chabane Djeraba. Sans sa rencontre il y a maintenant 5 ans, je n'aurais pas découvert les joies de la recherche et ne serais ainsi pas ici aujourd'hui. Je tiens également à remercier Christophe Chaillou pour m'avoir accueilli dans son équipe et pour s'être battu avec les démarches administratives.

Ce mémoire est l'aboutissement de trois années de travail menées au sein de l'équipe GRAPHIX, désormais nommée MINT. Je tiens à remercier l'ensemble des personnes qui ont contribué à la réalisation de ce travail.

Ensuite, mes remerciements vont à Géry Casiez et Laurent Grisoni. Leurs écoutes, leurs confiances ainsi que leurs esprits critiques ont été nombre d'éléments moteurs dans la bonne conduite de ces travaux. Ils ont su trouver la bonne méthode pour communiquer, m'accorder la liberté nécessaire, tout en étant disponible lorsque nécessaire. Les moments passés à leurs côtés pendant ces trois années m'ont beaucoup apporté. Merci.

Plus généralement, j'adresse un grand merci aux membres de l'équipe MINT, particulièrement Damien Marchal et Nicolas Roussel. Leurs conseils, leurs points de vue parfois différents ainsi que leurs analyses ont été d'autant plus d'élément m'emmenant vers la bonne voie.

Je n'oublie pas les collègues du deuxième étage de l'IRCICA qui m'ont apporté à un moment ou à un autre un coup de main, physique ou moral, pour mener à terme ces travaux. Spéciale dédicasse aux collègues du RU, Frédo, Gorio, Marie et Patrick, pour m'avoir rassuré et réconforté lorsque c'était nécessaire.

Je tiens également à remercier tous les proches qui m'ont entendu m'extasier, me morfondre, m'exciter, bref vivre ma thèse aux cours de ces trois années. Mention toute particulière à Marion qui a toujours (et continue de) trouvé les mots justes pour me faire avancer dans la bonne direction.

Finalement, je voudrais vous remercier vous, lecteur, pour prendre le temps de lire ces quelques lignes. J'espère que le reste de la lecture vous satisfera.

Table des matières

Introduction **1**

1 Motivations . . . 1
2 Contributions . . . 2
3 Plan du mémoire . . . 2
4 Terminologie . . . 3

Chapitre 1
État de l'art

1.1 Introduction . . . 5
1.2 Techniques de manipulation 3-D . . . 6
1.2.1 Techniques de manipulation 3-D à l'aide d'interfaces 2-D . . . 6
Positionnement 3-D contraint . . . 7
Positionnement 3-D libre . . . 7
Rotations 3-D . . . 10
1.2.2 Techniques de manipulation 3-D en réalité virtuelle . . . 10
Manipulation égocentrique . . . 11
Manipulation éxocentrique . . . 12
1.2.3 Bilan . . . 13
1.3 Les périphériques d'entrée multipoints . . . 13
1.3.1 Historique . . . 13
1.3.2 Les technologies tactiles . . . 16
La technologie acoustique . . . 16
La technologie électronique . . . 17
La technologie optique . . . 18
Avantages - Inconvénients . . . 20
1.3.3 Interaction 2-D et écrans multipoints . . . 20
Précision . . . 21

Manipulation 2-D 22
1.4 Techniques de manipulation 3-D multipoints 24
1.4.1 Techniques de manipulation indirecte 25
1.4.2 Techniques de manipulation directe 25
1.5 Intégration / séparation des degrés de liberté 28
1.5.1 Taxonomies 28
1.5.2 Structure de perception et structure de contrôle 28
1.5.3 Théories pour les interactions 3-D et la technologie multipoint . . . 29
1.6 Mesurer la coordination 31
1.6.1 Intégralité 31
1.6.2 Efficacité 32
1.6.3 Coefficient de Translation/Rotation 33
1.6.4 ***m**-metric* 34
1.6.5 Nombre de degrés de liberté combinés 34
1.7 Conclusion 35

Chapitre 2
Taxonomie des techniques d'interaction 3-D pour écrans multipoints

2.1 Introduction 37
2.2 État de l'art 38
2.3 Notre démarche 40
2.3.1 Constat 40
2.3.2 Proposition d'une taxonomie 42
Identification du nombre de contacts 42
Directivité 42
Manipulation intégrale ou séparable 43
2.4 Présentation de la taxonomie 43
2.4.1 Gestion des critères 43
Identification du nombre de contacts 43
Directivité 43
Manipulation intégrale ou séparable 44
2.4.2 Illustration des techniques existantes 44
Interaction *RST* 44
Interaction *Shallow-Depth* 44
Interaction avec moteur physique 45

2.5 Discussion et conclusion 45
2.5.1 Extensions possibles 45
Ajout de degrés de liberté 46
Adaptation aux périphériques indirects 46
Utilisation de la boîte englobante 47
2.5.2 Limites 48
2.5.3 Conclusion 48

Chapitre 3
Le positionnement d'objets 3-D

3.1 Introduction 49
3.2 Notre démarche 50
3.2.1 Technique 1 : *MT-viewport* 51
Viewport traditionnel 51
Viewports multipoints 52
3.2.2 Technique 2 : *Z-technique* 53
Translation à deux degrés de liberté 54
Translation à trois degrés de liberté 54
3.3 Évaluation des techniques 55
3.3.1 Objectifs 55
3.3.2 Matériel 55
3.3.3 Tâche réalisée 56
3.3.4 Participants 56
3.3.5 Plan expérimental 56
3.4 Résultats et discussion 57
3.4.1 Temps de positionnement 57
3.4.2 Coordination 58
Temps passé dans les différents modes 59
Nombre de degrés de liberté manipulés 60
Identification des degrés de liberté manipulés séparément 61
Identification des degrés de liberté manipulés intégralement 62
Efficacité et trajectoire 3-D 63
3.4.3 Utilisation de l'extension multipoint de la technique *MT-viewport* . 66
3.4.4 Analyse qualitative 67
3.5 Conclusion 68

Chapitre 4
La manipulation d'objets 3-D

4.1 Introduction 72
4.2 Techniques de manipulation 3-D existantes 73
4.2.1 *Sticky Tools* 73
Mode $1d$ 73
Mode $2d$ 74
Autres modes 74
4.2.2 *Screen-Space* 74
4.3 Introduction de la technique *DS3* 75
4.3.1 Notre démarche 75
4.3.2 Description de la technique *DS3* 77
Mode $1d$ et $1d + 1i$ 77
Mode $2d$ et $2d + 1i$ 77
4.4 Évaluation préliminaire 77
4.4.1 Matériel 78
4.4.2 Tâche et participants 78
4.4.3 Résultats et discussion 79
4.5 Expérience contrôlée pour l'évaluation des techniques 80
4.5.1 Objectifs 80
4.5.2 Tâche réalisée 80
4.5.3 Participants 80
4.5.4 Plan expérimental 81
4.6 Résultats et analyse 81
4.6.1 Indicateurs de difficulté 82
Ratio d'essais terminés 82
Nombre d'appuis 82
4.6.2 Temps de réalisation de la tâche 82
4.6.3 Coordination 83
Temps passé dans les différents modes 84
Nombre de degrés de liberté manipulés 85
Identification des degrés de liberté manipulés séparément 87
Identification des degrés de liberté manipulés intégralement 88
Efficacité de translation 90

Efficacité de rotation 92
Relation entre efficacité de translation et efficacité de rotation . . . 93
4.6.4 Analyse qualitative 95
4.7 Discussion et conclusion 96
4.7.1 Séparation des degrés de liberté et performance 97
4.7.2 Stratégies de séparation des degrés de liberté 97
4.7.3 Stratégie d'interaction bimanuelle 98
4.7.4 Contrôle direct vs. contrôle indirect 98

Conclusion **101**

Annexes **103**

Annexe A Trajectoires des techniques de positionnement 3-D **103**

Bibliographie **107**

Table des figures

1.1 Système de coordonnées utilisé par Nielson et Olsen [80] pour déplacer un objet 3-D en utilisant la position de la souris. 8
1.2 Les différents widgets 3-D utilisés dans Blender. 9
1.3 Capture d'écran du logiciel Maya illustrant la division de l'écran en plusieurs vues. 9
1.4 Illustration de la technique de pointeur virtuel *put-that-there* [13, 14]. . . . 11
1.5 Illustration de la technique *WIM*. 12
1.6 Photos de différents périphériques tactiles. 14
1.7 Photos de différents périphériques tactiles. 15
1.8 Illustration du geste *pinch-to-zoom*. 15
1.9 Illustration de différentes technologies tactiles. 16
1.10 Illustration du la technologie tactile capacitive. 17
1.11 Illustrations de la technologie tactile capacitive. 18
1.12 Illustrations des technologies tactiles optiques. 19
1.13 Illustration des techniques *TapTap* et *MagStick* [90]. 21
1.14 Illustration de la technique *Rubbing and Tapping* [83]. 22
1.15 Illustration des techniques de la famille *Cyclo** [68] : *CycloPan* et *Cyclo-Zoom+*. 22
1.16 Illustration du geste *RST* : Rotate-Scale-Translation. 22
1.17 Illustration de la technique *RNT* [63]. 23
1.18 Les différents curseurs introduits par Moscovich [78]. 24
1.19 Illustration de la technique *Shallow-Depth* [43]. 26
1.20 Illustration de la technique utilisant un moteur physique [113]. 26
1.21 Illustration de la technique *tBox* [35]. 27
1.22 Illustration de la notion d'intégrabilité introduite par Jacob et al. 29
1.23 Illustrations fictives des métriques de Zhai et al. 32

2.1 Illustration dans la taxomie de Card et l. [23] d'un écran et d'un pavé tactile contrôlant un pointeur de souris à l'écran. Un second contact a été ajouté au pavé tactile pour représenter l'opérateur d'agencement. 39
2.2 Première variante de l'illustration des limites de la taxonomie de Card et al. [23] en représentant le mécanisme d'interaction *RST*. Le deuxième contact est relié ici par l'opérateur d'agencement qui laisse une ambiguïté sur le contrôle des quatre degrés de liberté Tx, Ty, Tz et Rz. 41

2.3 Deuxième variante de l'illustration des limites de la taxonomie de Card et al. [23] en représentant le mécanisme d'interaction *RST*. Le deuxième contact est relié ici par l'opérateur d'agencement qui laisse une ambiguïté sur le contrôle des degrés de liberté Tz et Rz. 41
2.4 Troisième variante de l'illustration des limites de la taxonomie de Card et al. [23] en représentant le mécanisme d'interaction *RST*. Le deuxième contact est relié ici par l'opérateur de fusion qui laisse une ambiguïté sur le contrôle de Tx et Ty. 42
2.5 Illustration du mécanisme d'interaction *RST* dans notre taxonomie. 44
2.6 Illustration de la technique *Shallow-Depth* [43] dans notre taxonomie. . . . 45
2.7 Illustration de la technique de Wilson et al. [113] dans notre taxonomie. . . 45
2.8 Extension de notre taxonomie pour gérer plus de degrés de liberté. 46
2.9 Extension de notre taxonomie aux périphériques multipoints indirects : illustration du mécanisme *RST* indirect. 47
2.10 Illustration du concept de boîte englobante. 47
2.11 Illustration de *tBox* [35] dans notre taxonomie étendue. 48

3.1 Capture d'écran du logiciel Blender illustrant la gamme de technique divisant l'écran en plusieurs vues. 50
3.2 Illustration de la technique *MT-viewport* dans notre taxonomie. 52
3.3 Capture d'écran de la technique *MT-viewport*. Le doigt de gauche est utilisé pour un positionnement 2-D dans la plan de caméra correspondant au viewport associé. Le doigt de droite est utilisé pour téléporter le doigt et contrôler le troisième degré de liberté. La ligne grise représente le retour visuel indiquant à l'utilisateur le mouvement autorisé par le deuxième doigt. 53
3.4 Illustration de la technique *Z-technique* dans notre taxonomie. 53
3.5 Capture d'écran de la *Z-technique*. Le doigt de droite est utilisé pour un positionnement de manière directe dans le plan de la caméra tandis que le doigt de gauche est utilisé pour un positionnement indirect de la profondeur. Les mouvements en avant et en arrière rapprochent ou éloignent l'objet de l'utilisateur. 54
3.6 Temps de positionnement moyen pour chaque technique et précision. Les barres d'erreur représentent l'intervalle de confiance à 95%. 58
3.7 $\boldsymbol{m}$-metric moyenne pour chaque technique. Les barres d'erreur représentent l'intervalle de confiance à 95%. 63
3.8 Efficacité moyenne pour chaque technique et précision. Les barres d'erreur représentent l'intervalle de confiance à 95%. 64
3.9 Trajectoires 2-D et 3-D des différents participants pour la technique *MT-viewport*. 65
3.10 Trajectoires 2-D des différents participants pour la technique *Z-technique*. 66
3.11 Fréquence d'utilisation d'un second viewport. Les barres d'erreur représentent l'intervalle de confiance à 95%. 67

4.1 Illustration de la technique *Sticky Tools* dans notre taxonomie. 74
4.2 Illustration de la technique *Screen-Space* dans notre taxonomie. 75

4.3 Illustration de la technique *DS3* dans notre taxonomie. 76
4.4 Capture d'écran de la tâche *peg-in-hole*. 78
4.5 Valeurs moyennes pour la $\boldsymbol{m}$-metric et les différents couples étudiés. Nous constatons la très forte domination du couple (Tx, Ty). 88
4.6 Efficacité de translation moyenne pour chaque technique dans les différents niveaux de PROFONDEUR. Les barres d'erreur représentent l'intervalle de confiance à 95%. 91
4.7 Efficacité de rotation moyenne pour chaque technique dans les différents niveaux de PROFONDEUR. Les barres d'erreur représentent l'intervalle de confiance à 95%. 93
4.8 Ratio de translation-rotation moyen pour chaque technique. Les courbes débutent aux coordonnées $(1, 1)$ et terminent aux coordonnées $(0, 0)$. Le paramètre temps n'est pas représenté. 94
4.9 Exemples de courbes typiques de différents participants pour chaque technique. Illustration du contrôle simultanée avec les techniques *Sticky Tools* et *Screen-Space* et de la séparation avec la technique *DS3*. 95
4.10 Illustration du geste à 4 doigts avec la technique *Screen-Space* de Reisman et al. [88]. 96

A.1 Courbe de trajectoire 3-D pour la technique *Z-technique*. 104
A.2 Courbe de trajectoire 3-D pour la technique *MT-viewport*. 104
A.3 Courbe de trajectoire 2-D pour la technique *Z-technique*. 105
A.4 Courbe de trajectoire 2-D pour la technique *MT-viewport*. 106

Liste des tableaux

1.1 Tableau illustrant les différences entre les technologies tactiles. Le nombre d'étoiles indique la performance selon le critère. Plus le nombre d'étoiles est élevé, plus les performances du critère sont bonnes. 20

3.1 Pourcentage de temps passé dans les différents modes. 59
3.2 Résultat de la *NDC* pour les différentes techniques en pourcentage du temps global. 60
3.3 Pourcentage de temps passé à manipuler un degré de liberté unique. 61
3.4 Les différents viewports et les degrés de liberté associés pour la technique *MT-viewport*. 62

4.1 Pourcentage de temps passé dans les différents modes. 84
4.2 Résultat de la *NDC* pour les différentes techniques en pourcentage du temps global. 86
4.3 Pourcentage de temps passé à manipuler un degré de liberté unique. 87
4.4 Pourcentage de temps passé à manipuler intégralement des couples de degrés de liberté. 89

Introduction

1 Motivations

Le systèmes multipoints suscitent à l'heure actuelle un énorme intérêt que ce soit de la part du grand public ou encore des entreprises. En plus de permettre la consultation d'un contenu informatique, ce type de système permet également de toucher un élément du contenu afin de le sélectionner ou de déclencher une action, constituant ainsi un atout en grande partie responsable du succès des technologies multipoints et plus généralement tactiles. Pour des contenus informatiques simples, nous assistons d'ores et déjà à la mise en place d'applications au sein d'espaces publics tel que les musées ou les centres commerciaux. L'un des avantages pratiques de ce type d'environnements est d'instancier un autre relationnel, voulu plus spontané, entre l'utilisateur et la machine [49]. Potentiellement, tous les applicatifs où le clavier et la souris nécessitent un apprentissage sont autant de candidats éventuels à une réinvention, dans le cadre d'utilisation d'écrans multipoints, particulièrement adaptés à un contexte public [49]. Si utiliser une souris et un clavier nécessite un apprentissage, toucher pour interagir fait parti de l'apprentissage chez l'enfant, ceci dès son plus jeune âge.

Parallèlement à ce constat, de plus en plus de contenus sont aujourd'hui proposés en trois-dimensions (3-D). Ainsi, les techniques pour la 3-D interactive, issues du jeu vidéo, commencent à se répandre dans d'autres domaines comme le *serious game* pour la vente ou la rééducation, la simulation chirurgicale ou encore l'assemblage de molécules pour la chimie ou la biologie. Dans le contexte des systèmes multipoints, afin que de tels contenus puissent être utilisés de manière intéressante, se pose la question de savoir quelles techniques d'interaction employer. Dans le domaine de la réalité virtuelle, un certain nombre d'interactions élémentaires ont été historiquement identifiées [17], et la littérature est à l'heure actuelle riche de propositions de techniques d'interaction pour les diverses opérations élémentaires. De retour aux systèmes tactiles, l'enjeu important consiste à proposer des techniques d'interaction multipoints qui puissent, pour les contextes d'interaction précédemment mentionnés, proposer des outils adaptés à l'utilisateur ainsi qu'à son expertise.

Un des premiers exemples qui vient en tête dans ce contexte est celui de la manipulation d'objets 3-D, qui permet de définir les positions et orientations d'un objet. Il s'agit d'une des opérations les plus élémentaires que nous puissions envisager pour interagir avec du contenu 3-D. Dans le contexte infographique standard, les outils comme Maya, 3DS ou Rhino proposent des interfaces similaires (cf chapitre 1), assez denses et techniques, permettant à l'utilisateur de positionner très finement des objets 3-D. Ces interfaces nécessitent toutefois un apprentissage important, et ne sont pas adaptées dans le contexte

grand public cité plus haut. Il s'agit donc d'envisager une autre voie de manipulation, probablement moins précise mais qui serait adaptée au contexte utilisateur grand public.

Ce mémoire s'intéresse donc à l'une des techniques d'interaction 3-D les plus élémentaires, la manipulation d'objets, sur dispositif multipoint. Les travaux réalisés s'inscrivent également dans le projet de recherche européen (ANR) *InSTInCT*[1] consacré à concevoir, développer et évaluer de nouvelles interfaces utilisateur basées sur des surfaces tactiles pour favoriser l'interaction 3-D, notamment pour des applications 3-D à destination du grand public.

2 Contributions

Après une présentation de l'état de l'art, les contributions de ce mémoire s'organisent en trois parties.

Tout d'abord nous introduirons une taxonomie afin de représenter les techniques d'interaction de manipulation 3-D propres aux écrans multipoints. Cette taxonomie nous permettra de visualiser rapidement les différences de conception entre les différentes techniques d'interaction, de manière à identifier les facteurs qui peuvent influencer la réalisation de la tâche.

Ensuite, nous nous intéresserons à une sous tâche de la manipulation 3-D : le positionnement 3-D. Nous effectuerons ainsi une étude des techniques d'interaction permettant de positionner un objet 3-D à l'aide de surfaces multipoints. Nous introduirons deux techniques propres aux écrans multipoints et présenterons des résultats ainsi qu'une analyse quant à la réalisation de ce type de tâche par l'utilisateur.

Enfin, nous effectuerons une étude des techniques de manipulation 3-D, c'est à dire les techniques permettant de gérer à la fois la position et l'orientation. Nous présenterons ainsi une nouvelle technique basée sur l'analyse de l'existant et étudierons avec précision la coordination dans la réalisation de la tâche par l'utilisateur.

Ces différentes contributions ont été publiées dans les articles suivants : [71, 73, 69, 72, 70]

3 Plan du mémoire

La structure du mémoire s'appuie sur l'ordre des contributions précédemment énumérées.

Dans un premier chapitre, nous présenterons un état de l'art orienté autour de plusieurs axes. Tout d'abord nous rappellerons les techniques de manipulation 3-D existantes autant pour les interfaces de bureau 2-D que pour la réalité virtuelle. Puis, nous effectuerons un historique de la technologie multipoint et présenterons les techniques d'interaction 2-D et 3-D existantes. Ensuite nous nous concentrerons sur les théories concernant l'intégration et la séparation des degrés de liberté pour la réalisation d'une tâche 3-D à l'aide de surfaces multipoints. Enfin, nous exposerons différentes métriques utilisées pour la mesure de la

1. `http://anr-instinct.cap-sciences.net/`

coordination, métriques qui nous permettront par la suite de caractériser les stratégies d'intégration et/ou de séparation.

Dans un second chapitre, nous introduirons une nouvelle taxonomie basée sur l'analyse des limites des taxonomies existantes. Nous détaillerons les étapes de conception et illustrerons cette nouvelle taxonomie à l'aide d'exemples.

Dans un troisième chapitre, nous effectuerons une étude des techniques de positionnement 3-D pour écrans multipoints. Nous présenterons ainsi deux techniques tirant profit des possibilités offertes par la technologie multipoint. Nous évaluerons ensuite ces deux techniques et orienterons nos résultats autour de la coordination dans le but d'identifier l'influence des stratégies de conception des techniques, notamment l'influence de la séparation des degrés de liberté.

Dans un dernier chapitre, nous effectuerons une étude des techniques de manipulation 3-D pour écrans multipoints. Nous analyserons les stratégies d'intégration ou de séparation des degrés de liberté pour les techniques existantes et introduirons une nouvelle technique basée sur cette analyse. Enfin, nous évaluerons les différentes techniques afin de caractériser les conséquences des différentes stratégies sur la réalisation de la tâche, principalement en terme de coordination.

4 Terminologie

Quelques termes se doivent d'être définis afin de pouvoir éclairer le lecteur sur la problématique de ce mémoire. Comme explicité dans les motivations, nous réfèrerons au domaine à deux dimensions (respectivement trois dimensions) par l'abréviation 2-D (respectivement 3-D).

La manipulation 3-D constitue le cœur de ce mémoire. Par manipulation 3-D, nous entendons la gestion de la position ainsi que la gestion de l'orientation.

Afin de réaliser un positionnement 3-D, nous devons gérer trois degrés de liberté, afin de pouvoir effectuer des déplacements sur les trois axes d'un environnement 3-D : les axes x, y et z. Dans la suite de ce mémoire nous réfèrerons à ces degrés de liberté de position par les dénominations Tx, Ty et Tz (le T indiquant une translation). De manière similaire, l'orientation 3-D requiert la gestion de trois degrés de liberté que nous nommerons Rx, Ry et Rz (le R indiquant une rotation).

Le repère géométrique utilisé tout au long de ce mémoire est un repère cartésien direct où l'axe x est situé dans le plan de l'écran, orienté vers la droite et où l'axe z est orthogonal au plan de l'écran, pointant vers l'utilisateur.

Chapitre 1

État de l'art

Sommaire

1.1 Introduction 5
1.2 Techniques de manipulation 3-D 6
1.2.1 Techniques de manipulation 3-D à l'aide d'interfaces 2-D . . . 6
1.2.2 Techniques de manipulation 3-D en réalité virtuelle 10
1.2.3 Bilan 13
1.3 Les périphériques d'entrée multipoints 13
1.3.1 Historique 13
1.3.2 Les technologies tactiles 16
1.3.3 Interaction 2-D et écrans multipoints 20
1.4 Techniques de manipulation 3-D multipoints 24
1.4.1 Techniques de manipulation indirecte 25
1.4.2 Techniques de manipulation directe 25
1.5 Intégration / séparation des degrés de liberté 28
1.5.1 Taxonomies 28
1.5.2 Structure de perception et structure de contrôle 28
1.5.3 Théories pour les interactions 3-D et la technologie multipoint 29
1.6 Mesurer la coordination 31
1.6.1 Intégralité 31
1.6.2 Efficacité 32
1.6.3 Coefficient de Translation/Rotation 33
1.6.4 ***m**-metric* 34
1.6.5 Nombre de degrés de liberté combinés 34
1.7 Conclusion 35

1.1 Introduction

Traditionnellement, les techniques d'interaction pour les tâches 3-D peuvent être divisées en plusieurs catégories [17]. La première catégorie regroupe les techniques relatives

à la navigation 3-D, c'est à dire les tâches qui nécessitent la manipulation d'une caméra virtuelle afin de se déplacer (*travel*) et/ou trouver un chemin (*wayfinding*) dans un univers 3-D. La seconde catégorie concerne les techniques de manipulation 3-D, techniques qui consistent à sélectionner, positionner puis orienter des objets 3-D dans un univers virtuel. Deux autres catégories existent mais restent moins étudiées que les deux premières : les tâches permettant le contrôle d'un système, c'est à dire les tâches visant à changer l'état de celui-ci par l'intermédiaire de commandes ou menus, et les tâches de saisies de symboles (textes, nombres). Dans ce mémoire, nous allons nous intéresser particulièrement aux tâches de manipulation 3-D. Notre état de l'art commencera donc par une présentation des techniques d'interaction existantes permettant d'effectuer des tâches de manipulation 3-D, que ce soit avec des interfaces 2-D traditionnelles comme la souris ou en réalité virtuelle.

Nous consacrerons le deuxième axe de l'état de l'art aux périphériques d'entrée multipoints, périphériques situés au cœur de ce mémoire. Nous présenterons ainsi un historique, détaillerons les différentes technologies existantes en mettant en avant leurs forces ainsi que leurs faiblesses puis terminerons en exposant différentes techniques d'interaction 2-D que nous jugerons pertinentes.

Ces deux premiers axes seront ensuite reliés dans une troisième section qui sera consacrée à la présentation de techniques de manipulation 3-D spécifiques aux écrans multipoints.

Nous verrons ensuite que l'étude des techniques existantes soulèvent des questions relatives à l'intégration ou à la séparation des degrés de liberté. Nous aborderons cet aspect dans une quatrième section qui portera la réflexion sur les théories relatives aux stratégies d'intégration, que ce soit de manière générale ou spécifique aux périphériques multipoints.

Enfin, nous présenterons les métriques utilisées pour mesurer l'intégration ou la séparation des degrés de liberté, et plus généralement la coordination lors de la réalisation d'une tâche 3-D.

1.2 Techniques de manipulation 3-D

Nous allons commencer l'état de l'art en présentant les techniques d'interactions permettant la réalisation de tâches de manipulation 3-D. Pour cela, nous allons dans un premier temps présenter les techniques d'interaction utilisant comme périphérique d'entrée la souris. Ensuite, nous nous attarderons à présenter les techniques traditionnellement employées dans le domaine de la réalité virtuelle, utilisant généralement des périphériques d'entrée à six degrés de liberté.

1.2.1 Techniques de manipulation 3-D à l'aide d'interfaces 2-D

La manipulation 3-D requiert à la fois la gestion de la position et de l'orientation. Dans le cas d'utilisation d'interfaces 2-D avec des périphériques tels que la souris, nous avons remarqué que la gestion des attributs de position et d'orientation sont généralement traitées de manière distincte. C'est pourquoi nous avons choisi de catégoriser les techniques

d'interaction en séparant les techniques de positionnement des techniques de gestion de l'orientation. Dans un premier temps, nous présenterons les techniques permettant d'effectuer ce que nous qualifierons de positionnement *contraint* : l'utilisateur manipule un sous-ensemble de degrés de liberté, choisi automatiquement par la technique en utilisant des informations de la scène. Ensuite nous présenterons les techniques permettant d'effectuer un positionnement libre, où l'utilisateur contrôle lui-même les degrés de liberté. Enfin, nous présenterons les techniques de gestion de l'orientation.

Positionnement 3-D contraint

Le positionnement contraint repose sur l'idée de trouver certaines propriétés d'une scène 3-D afin de faciliter la manipulation d'un objet dans celle-ci.

Bier fut un des pionniers du domaine et proposa des techniques d'interaction pour la souris. En 1986, il introduit le concept de *snap-dragging* [10]. A l'aide des propriétés visuelles d'une scène 2-D, il propose de faciliter le déplacement en effectuant une accroche lors du déplacement. Cette rupture va déplacer l'objet manipulé sur une propriété visuelle située à proximité de l'objet. L'option "aligner sur la grille", omniprésente dans les logiciels de dessins actuels, est inspirée de ces travaux. Initialement présentée dans un contexte à deux dimensions, Bier présentera plus tard une version similaire, appliquée à la 3-D [9]. Cinq ans plus tard, Bukowski et al. [18] ont introduit une technique qui scrute l'espace 3-D situé *sous* la position du curseur afin de contraindre le positionnement de l'objet manipulé. Cette technique permet de grouper et de manipuler plusieurs objets, simplifiant ainsi la construction de scènes 3-D. Toujours dans la même idée, Oh et al. [82] ont amélioré le *snap-dragging* de Bier. Tandis que ce dernier utilisait uniquement la position du curseur comme entrée pour son algorithme, Oh et al. ont augmenté cette zone en incluant l'entière surface présente derrière l'objet pour la recherche de propriétés visuelles afin de déterminer le déplacement à effectuer. Ils réalisèrent une étude expérimentale comparant cette technique avec la technique des ancres présente à l'époque dans l'application Maya [2]. Cette étude révéla une supériorité de leur technique concernant la rapidité de réalisation de la tâche.

Pour toutes ces techniques, les degrés de liberté manipulés sont choisis automatiquement par l'application en fonction de l'environnement. Malgré leurs présences dans de nombreuses solutions commerciales, ces techniques de manipulation possèdent des propriétés intrinsèques, rendant leur utilisation parfois difficile. Par exemple, dans le cas de scènes 3-D complexes où de nombreux objets sont présents, contraindre l'objet à se positionner sur des propriétés visuelles résulte en de petits sauts discontinus, rendant l'interaction difficile. Lorsque la scène manipulée ne possède pas de sol, l'usage de telles techniques est également réduit. Les techniques de positionnement 3-D libre représentent une alternative viable, permettant de contrôler librement les objets d'une scène 3-D, quelque soit la complexité de la scène ou les relations entre les objets.

Positionnement 3-D libre

Nielson et Olsen [80] ont proposé les premiers une technique permettant de contrôler une position 3-D à l'aide de périphériques d'entrée possédant deux degrés de liberté (*e.g.*

2. `http://www.autodesk.fr/adsk/servlet/pc/index?siteID=458335&id=14657544`

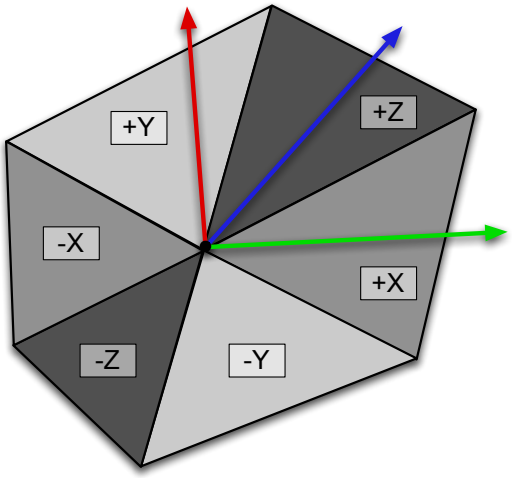


FIGURE 1.1 – Système de coordonnées utilisé par Nielson et Olsen [80] pour déplacer un objet 3-D en utilisant la position de la souris.

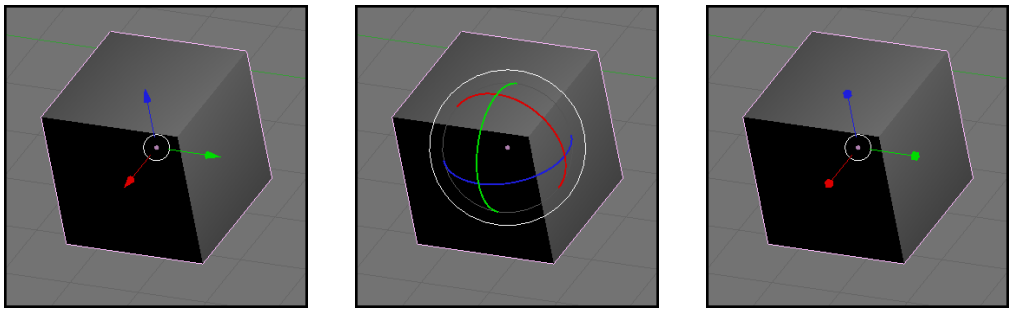
la souris). Leur technique contraint la translation à un axe unique, fonction du plus grand déplacement dans un des axes du système de coordonnées (figure 1.1). Cette idée est encore présente aujourd'hui dans des solutions commerciales, comme par exemple Google Sketchup [3]. Contrairement aux techniques de manipulation contrainte où l'application restreignait les degrés de liberté, c'est ici l'utilisateur qui choisi comment séparer les degrés de liberté. Cette technique peut cependant poser des problèmes pour les petits mouvements précis, où la prédiction peut ne pas refléter le choix de l'utilisateur.

Bier [11] a imaginé une solution plus générale. Sa technique consiste à attacher des systèmes de coordonnées différents à chaque objet 3-D. Ce système de coordonnées permet de sélectionner chacun des axes séparément afin de contrôler la position de l'objet. La technique possède l'avantage de pouvoir effectuer des actions précises mais peut cependant nécessiter de nombreuses actions successives, rendant la réalisation de la tâche laborieuse. Ce genre de technique impose également une importante charge cognitive, l'utilisateur devant décomposer mentalement les opérations à effectuer.

Toujours dans une approche où les degrés de liberté sont contrôlés séparément par l'utilisateur, Venolia [103] propose une technique qui se veut *facile* d'utilisation. Il utilise la position 2-D de la souris pour effectuer le positionnement dans le plan de caméra (manipulation de Tx et Ty) et utilise la molette pour contrôler séparément le déplacement en profondeur (Tz). Plus globalement, l'utilisation de la molette pour contrôler le zoom est aujourd'hui courant dans les applications 3-D, comme la Conception Assistée par Ordinateur (CAO).

Les widgets 3-D et autres ancres [36, 95] représentent une alternative permettant également de manipuler librement un objet 3-D (voir figure 1.2). Ces techniques sont aussi toutes basées sur la séparation des degrés de liberté. Contrairement aux techniques de positionnement 3-D contraint où l'application restreignait le déplacement, c'est ici l'utilisateur qui fait le choix d'activer un widget, sélectionnant ainsi un sous-ensemble

3. `http://sketchup.google.com/intl/fr/`



(a) Widget de translation (b) Widget de rotation (c) Widget de mise à l'échelle

FIGURE 1.2 – Les différents widgets 3-D utilisés dans Blender.

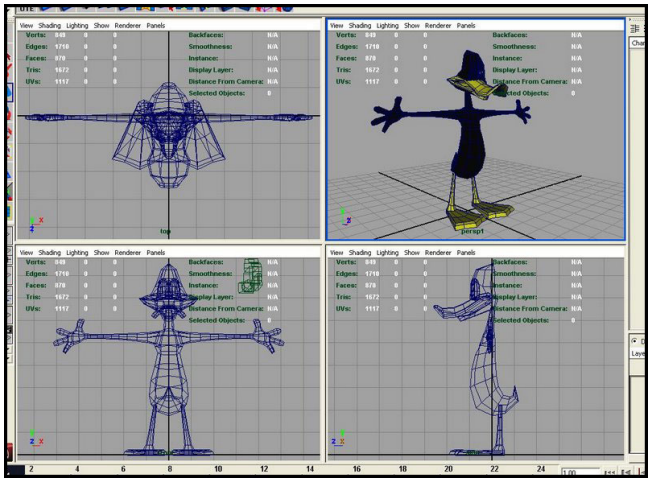


FIGURE 1.3 – Capture d'écran du logiciel Maya illustrant la division de l'écran en plusieurs vues.

de degrés de liberté à manipuler. Par exemple, dans la figure 1.2a, si l'utilisateur clique sur l'axe vert, il va pouvoir effectuer des translations uniquement suivant cet axe. Les widgets 3-D sont utilisés dans de nombreuses applications commerciales actuelles (*e.g.* Maya, Blender, Unity3D).

L'autre solution utilisée aujourd'hui massivement dans les logiciels de 3-D consiste à diviser l'écran en plusieurs vues de la scène 3-D. Traditionnellement, le nombre de vues était de quatre comprenant trois vues orthogonales et une vue perspective de la scène 3-D (figure 1.3). Quand l'utilisateur manipule un objet 3-D dans une vue orthogonale, le déplacement possible est réduit à deux degrés de liberté, définis par le plan de la vue orthogonale. Cette technique a été démontrée plus rapide que les techniques plein-écran utilisant des ancres pour une tâche de positionnement 3-D [82]. Notons également qu'il est courant de trouver une combinaison de techniques comprenant à la fois plusieurs vues de la scène et des widgets 3-D.

Nous venons de présenter les techniques permettant la gestion de la position d'un objet 3-D à l'aide de la souris, passons maintenant à la gestion de l'orientation.

Rotations 3-D

Gérer les degrés de liberté responsables de l'orientation 3-D d'un objet est une tâche difficile [92]. Pour faciliter ce processus, une grande famille de techniques d'interaction existe : les *trackballs*.

Tout d'abord, nous trouvons les techniques dîtes de "sphère virtuelle" comme la *Virtual Sphere* [34], la *Rolling Ball* [46] ou encore la *Virtual Trackball* [51]. L'idée derrière ces techniques est simple : imaginez que l'objet 3-D que vous souhaitez orienter est enfermé dans un globe de verre qui peut pivoter librement autour de son centre. Alors, au lieu d'orienter directement l'objet, il suffit de pousser avec le curseur de la souris le globe de verre qui pivotera autour de l'axe perpendiculaire au mouvement du curseur. Pour faciliter la visualisation pour cette technique, un cercle ou une sphère sont généralement représentés autour de l'objet à orienter. Ce genre de technique ne permet cependant pas de rotation directe autour de l'axe pointant vers l'écran (*i.e.* perpendiculaire à l'écran). Pour palier à ce manque, les mouvements du curseur situés en dehors de la sphère de visualisation sont parfois assignés à cette tâche.

Cependant, afin de présenter un modèle complet autorisant les rotations autour de l'axe perpendiculaire à l'écran sans passer par un contrôle indirect, une autre métaphore a été créée : l'*ARCBALL* [93]. Pour visualiser le fonctionnement de cette technique, reprenons la précédente représentation à l'aide du globe de verre. Cette fois-ci, imaginez que vous plantez une aiguille dans ce globe. Lorsque vous effectuez un mouvement à l'aide de la souris, vous déplacez l'aiguille sur la surface du globe, effectuant ainsi une rotation 3-D. Cette technique, qui utilise les quaternions pour modéliser les rotations, est aujourd'hui très utilisée.

Ainsi, la technique de gestion de l'orientation *ARCBALL*, alliée aux multiples vues ou aux widgets pour contrôler la position 3-D, constitue aujourd'hui la méthode de manipulation 3-D de bureau standard. Ces interactions permettent de simplifier la compréhension de l'utilisateur en décomposant les différentes actions à effectuer. Cependant, de nombreuses actions répétées sont parfois nécessaires rendant la tâche fastidieuse.

Passons maintenant aux techniques utilisées dans le domaine de la réalité virtuelle. Ces techniques tirent profit du grand nombre de degrés de liberté proposés par le périphérique d'entrée (souvent six) et quelquefois de la projection stéréoscopique de la scène 3-D.

1.2.2 Techniques de manipulation 3-D en réalité virtuelle

La réalité virtuelle nécessitant une visualisation et des périphériques d'entrée très éloignés des écrans multipoints, nous ne serons pas exhaustif et introduirons seulement les grandes familles de techniques d'interaction. Pour présenter ces techniques, nous allons reprendre la taxonomie proposée par Poupyrev et al. [85, 87]. Dans cette taxonomie, les techniques sont classifiées comme *égocentrique* ou *éxocentrique*. Dans les interactions éxocentriques, l'utilisateur interagit avec l'environnement 3-D en se situant en dehors de celui-ci. A contrario, dans les interactions égocentriques, l'utilisateur se situe dans l'environnement et interagit en vue à la première personne.

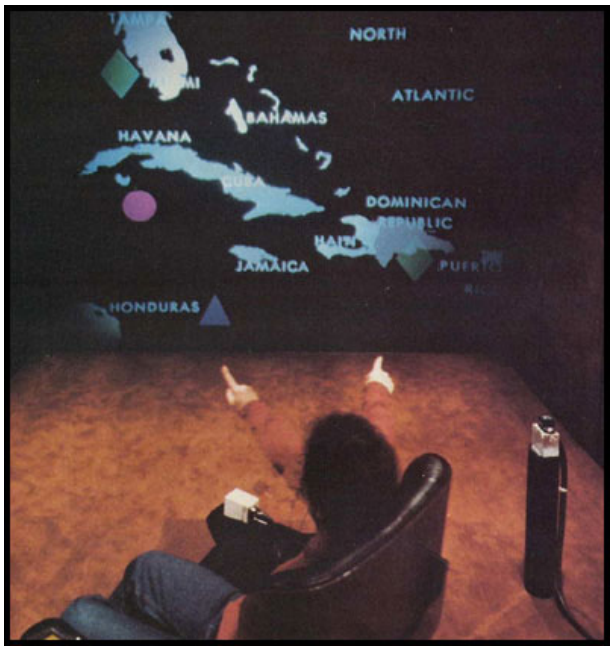


FIGURE 1.4 – Illustration de la technique de pointeur virtuel *put-that-there* [13, 14].

Manipulation égocentrique

Deux familles de techniques d'interactions égocentriques sont présentes.

Tout d'abord nous trouvons la famille de techniques appelée "pointeur virtuel" (*virtual pointer*). Ici, l'utilisateur interagit en pointant les objets qu'il souhaite manipuler. La motivation de telles techniques réside dans la facilité pour l'utilisateur à sélectionner et manipuler des objets situés loin de lui en les pointant. Lorsqu'un objet est sélectionné il devient alors attaché au pointeur, ce qui permet ensuite d'effectuer la manipulation. Une des premières implémentations fut le *met-ça-ici* (*put-that-there*), développé par Bolt dans les années 1980 [13, 14] (figure 1.4). Depuis plusieurs variantes ont été présentées [15, 56, 77]. Ces variantes diffèrent principalement sur deux aspects. Premièrement, la manière dont la direction de pointage est définie, puis la forme du rayon de sélection qui défini le retour visuel ainsi que le nombre d'objets sélectionnables par la technique. De manière générale, ces techniques sont très efficaces pour la sélection car elles nécessitent un faible mouvement de la part de l'utilisateur [16, 85]. Cependant, ces techniques rendent la manipulation difficile car seuls sont efficaces les mouvements radiaux autour de l'utilisateur (perpendiculaire à la direction de pointage). De même les rotations ne sont possibles qu'autour de l'axe défini par le pointeur. Pour palier à ce problème de manipulation, passons à la deuxième famille de techniques égocentriques.

L'autre famille de métaphore est appelée "main virtuelle" (*virtual hand*). Dans ce type d'interaction, l'utilisateur manipule les objets en les touchant à l'aide d'une main virtuelle. Cette main est contrôlée directement par le périphérique d'entrée qui possède le nombre de degrés de liberté requis pour manipuler à la fois la position et l'orientation. Il s'agit ici d'une interaction isomorphique : la position et l'orientation du périphérique sont liés à la main virtuelle. Les objets manipulables par l'utilisateur sont donc ceux qui se trouvent à

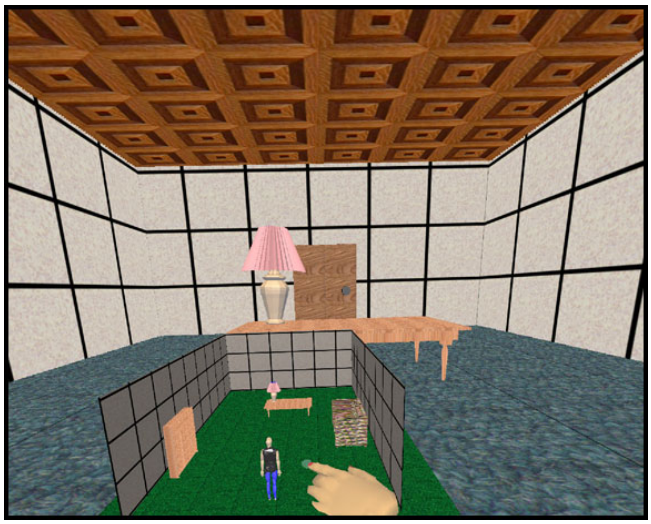


FIGURE 1.5 – Illustration de la technique *WIM*.

portée de lui. Cela pose un problème dans le cas où l'utilisateur souhaite manipuler des objets qui sont éloignés de lui dans la scène 3-D. Pour résoudre ce problème, Poupyrev et al. [86] ont présenté la *Go-Go* technique. Dans cette technique, la longueur du bras virtuel est modifiable en fonction de la distance réelle du bras de l'utilisateur avec son buste. Une fonction de transfert relie cette distance réelle avec la longueur du bras virtuel afin de pouvoir atteindre à la fois des objets proches et très éloignés [33, 27].

Nous venons de présenter deux familles d'interactions égocentriques pour la réalité virtuelle. Les techniques de pointeurs virtuels sont excellentes pour la sélection d'objets tandis que les techniques de mains virtuelles sont préconisées pour la manipulation. Passons maintenant à l'autre catégorie : les interactions éxocentriques.

Manipulation éxocentrique

Pour les techniques d'interaction éxocentriques, l'utilisateur se situe en dehors de l'environnement 3-D qu'il souhaite manipuler. La plus célèbre est la technique "Monde-en-Petit" ou *WIM* (*World-in-Miniature*) [94].

Cette technique résout le problème de longueur de bras que l'utilisateur peut rencontrer avec les techniques de main virtuelle. *WIM* permet à l'utilisateur de manipuler une copie *miniature* de l'environnement 3-D (figure 1.5). Cela permet de manipuler indirectement les objets en interagissant avec leurs copies dans le monde miniature. Cette technique est puissante car elle permet de manipuler des objets à la fois proches et éloignés de l'utilisateur. Cependant, cette technique supporte mal la mise à l'échelle. Très adéquate pour les environnements petits et moyens (pièces ou bâtiments), elle devient difficilement applicable pour les grands environnements où la manipulation de petits objets devient impossible de part le facteur d'échelle.

En conclusion, les techniques utilisées en réalité virtuelle peuvent paraître plus *naturelles* du fait des mouvements utilisés qui sont très proches de ceux de la vie quotidienne. Néanmoins, ces techniques restent très peu utilisées car les interfaces de réalité virtuelle

sont très chères et largement moins répandues que les interfaces de bureau utilisées quotidiennement par de nombreux professionnels. Dans ces interfaces de bureau, le périphérique d'entrée le plus utilisé est la souris qui permet le contrôle d'un seul et unique pointeur. Les interfaces multipoints apparaissent donc prometteuses car elles permettent de contrôler simultanément plusieurs pointeurs, augmentant la bande passante d'entrée du système. Récapitulons maintenant la partie sur la manipulation 3-D.

1.2.3 Bilan

Tout comme dans le monde réel où un seul outil ne peut réaliser toutes les tâches, il n'existe pas de techniques d'interaction qui soient universelles. Suivant la tâche à effectuer, il convient de sélectionner et de combiner les techniques qui sont les plus appropriées. Pour les interfaces 2-D de type bureau, la manipulation est ainsi généralement réalisée en combinant les techniques de rotations avec des widgets 3-D ou encore de multiple vues de la scène. Pour la réalité virtuelle, la sélection se fera usuellement avec une technique pointeur virtuel associée à une technique de main virtuelle pour la manipulation.

Comme nous pouvons le voir, les techniques dépendent grandement du périphérique d'entrée utilisé. Nous allons donc, dans la prochaine partie, nous intéresser aux périphériques d'entrée multipoints, afin de présenter leurs spécificités.

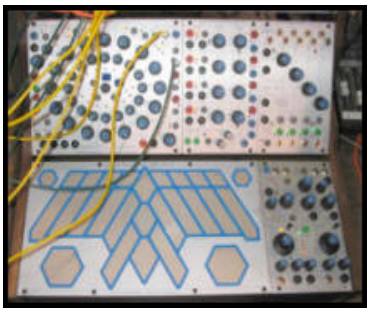
1.3 Les périphériques d'entrée multipoints

Dans cette section, nous allons présenter la technologie multipoint. Nous débuterons par un bref historique, expliquant les dates clefs de cette technologie arrivée aujourd'hui à maturité. Ensuite nous présenterons les différents méthodes utilisées pour permettre à une surface de détecter plusieurs contacts, en mettant en relief les avantages et inconvénients propres à chaque approche. Nous conclurons en présentant quelques techniques d'interaction 2-D spécifiques aux écrans multipoints, techniques que nous avons jugées pertinentes dans notre réflexion autour de la manipulation 3-D.

Avant de débuter l'historique, nous tenions à expliciter une notion très importante pour les périphériques tactiles : la directivité. Cette notion nous permet de catégoriser les différents périphériques en deux classes : direct ou indirect. Un périphérique tactile est dit direct si la surface d'entrée est confondue avec la surface de sortie. Les bornes tactiles présentes dans les gares françaises sont ainsi des exemples de périphériques directs. A contrario, les pavés tactiles des ordinateurs portables sont des périphériques indirects : la surface d'entrée et la surface d'affichage sont distinctes.

1.3.1 Historique

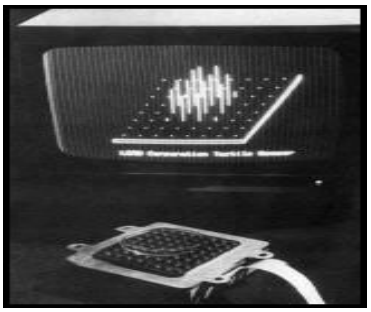
Cela peut sembler éloigné du sujet, mais le clavier fut le premier système permettant plusieurs entrées simultanées. Contrairement aux téléphones tactiles actuels, la détection est effectuée de manière mécanique : l'utilisateur doit appuyer sur différentes touches mécaniques. Néanmoins, il s'agit d'un système capable de détecter plusieurs entrées. Les



(a) Années 1960 : périphérique indirect destiné à la musique



(b) 1972 : Le *PLATO IV* : périphérique monopoint direct



(c) 1981 : Premier capteur indirect multipoint

FIGURE 1.6 – Photos de différents périphériques tactiles.

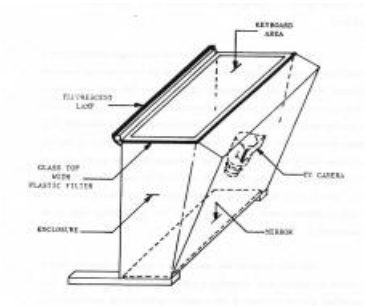
combinaisons utilisant les touches *Alt*, *Shift* ou *Ctrl* en sont le parfait exemple, avec pour illustration le désormais célèbre *Ctrl+Alt+Suppr*.

Le développement de systèmes non mécaniques, de périphériques *tactiles* commença à la fin des années 1960 (figure 1.6a). Les premiers travaux ont été réalisés par IBM, à l'université de l'Illinois et à Ottawa [4]. En 1971, plusieurs systèmes tactiles étaient opérationnels. Ces systèmes avaient deux points communs. Tout d'abord, tous étaient monopoint, détectant un seul contact. Ensuite, ces capteurs n'étaient pas capables de mesurer la pression exercée. Le premier périphérique tactile disponible pour le grand public fut le *PLATO IV* [5], déployé en 1972 (figure 1.6b). Deux innovations majeures se trouvaient dans ce produit : un écran tactile bien sûr, mais également un écran plasma, inédit à l'époque.

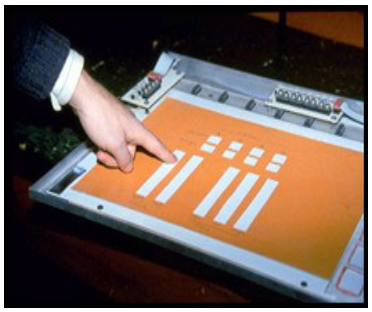
Le premier système de capteurs capable de détecter plusieurs points fut créé en 1981 par Jack Rebman (usage décrit dans [114]). Il s'agissait ici d'un périphérique indirect : l'affichage est déporté du système de commande. Ce système resta cantonné au domaine de l'électronique. Le premier périphérique d'entrée multipoint utilisé pour l'Interaction Homme Machine (IHM) fut développé par Nimish Mehta à l'université de Toronto en 1982 [76] (figure 1.7a). Il s'agissait d'un panneau en verre dépoli possédant certaines propriétés optiques. L'arrière de la surface était filmé par une caméra tandis que l'avant servait de surface d'interaction pour les utilisateurs. Lorsqu'un doigt se posait sur la surface, le fond initialement blanc laissait apercevoir des tâches noires captées ensuite par les caméras. Un algorithme traitait l'image issue de la caméra pour obtenir les points utilisés pour l'interaction, limitée à cette époque aux tâches de dessin. Ici aussi, il s'agissait d'un périphérique indirect. La méthode employée fut par la suite adaptée et nous retrouvons aujourd'hui la même approche pour les périphériques multipoints utilisant la technologique optique.

Le premier périphérique multipoint direct (nous qualifierons par la suite les périphériques multipoints directs d'écrans multipoints) fut développé par Bob Boie en 1984[4]. Le système utilisait une matrice de capteurs de contact transparents, superposée à un écran à tubes cathodiques. L'écran permettait de manipuler des objets graphiques avec un temps de réponse jusqu'ici inégalé. En 1985, Lee et al. [65] présentent une tablette (périphérique

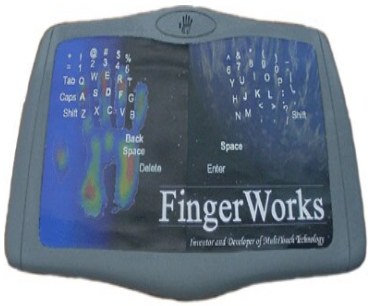
4. `http://www.billbuxton.com/multitouchOverview.html`
5. `http://en.wikipedia.org/wiki/Plato_computer`



(a) 1982 : *Flexible Machine Interface*



(b) 1985 : Lee et Buxton



(c) 1998 : pavé tactile FingerWorks

FIGURE 1.7 – Photos de différents périphériques tactiles.

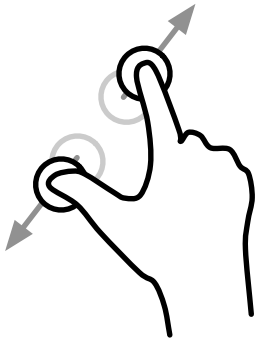
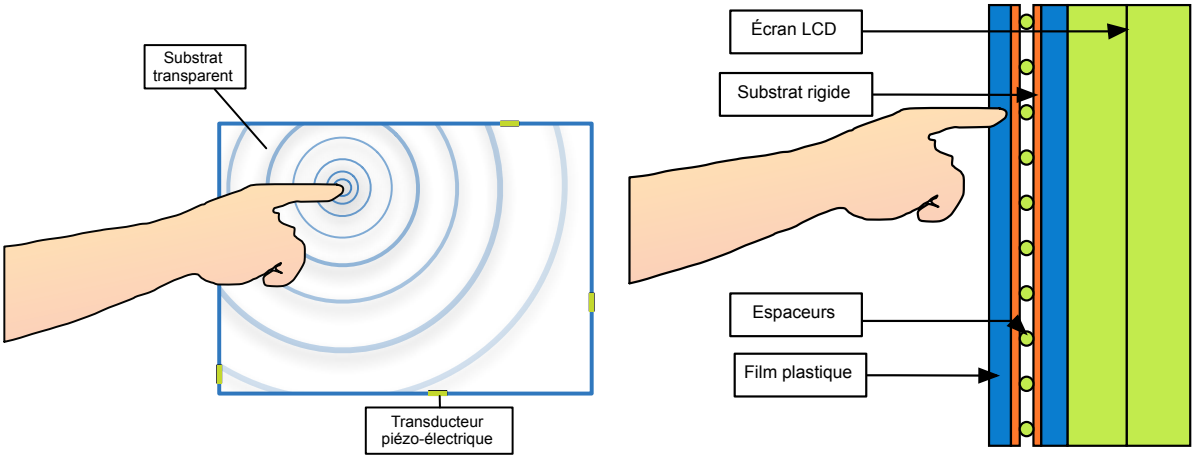


FIGURE 1.8 – Illustration du geste *pinch-to-zoom*.

indirect) capable de détecter la pression (figure 1.7b). Nombre de systèmes similaires apparurent dans les années 1980 et 1990. Cette époque fut également marquée par les premières réflexions sur les possibilités offertes par de tels dispositifs. Myron Krueger fut un pionner dans ce domaine. Il décrit dans ses travaux [59, 60, 61, 62] de l'époque le geste *pinch-to-zoom* —que nous pourrions traduire par *écarter pour agrandir*— aujourd'hui largement utilisé dans les produits commerciaux (figure 1.8).

Les années 2000 marquèrent l'essor de la technologie [37, 50, 89, 110, 111]. En 1998, suite à ses travaux de doctorat [109], Wayne Westerman co-fonde l'entreprise FingerWorks avec un collègue. Ensemble, ils produisent des pavés tactiles (figure 1.7c) capables de détecter plusieurs points. Leur société sera rachetée en 2005 par Apple. En 2001, Le laboratoire Mitsubishi à Cambridge dévoile *DiamondTouch* [37] qui offre la possibilité de collaboration entre utilisateurs, de fait de sa grande taille. Cette table possède une fonctionnalité unique : elle est capable d'identifier précisément les utilisateurs. En 2002, Jun Rekimito présente *SmartSkin* [89]. Ces deux prototypes souffrent néanmoins d'un problème de coût et de complexité de développement. Il faudra attendre 2005 pour que Han présente un système très bon marché [42], permettant de réaliser un écran multipoint efficace, facilement réalisable et peu coûteux. Ses travaux ont permis un développement très important, rendant la technologie accessible à de nombreux laboratoires. En 2007, Apple sort son iPhone[6], démocratisant la technologie auprès du grand public. En 2010,

6. `http://www.apple.com/fr/iphone/`



(a) Technologie acoutisque *APR*. (b) Technologie résistive.

FIGURE 1.9 – Illustration de différentes technologies tactiles.

l'organisme Garner [7] estime que la vente des téléphones tactiles dépassera 360 millions d'unités.

En résumé, nous pouvons voir que les périphériques multipoints ont subi un développement similaire à celui de la souris. Une trentaine d'années se sont ainsi écoulées pour passer des prototypes présents dans les laboratoires de recherche à la commercialisation et l'usage à grande échelle. La technologie est ainsi arrivée à maturité. Nous allons maintenant détailler les différentes approches utilisées pour réaliser des systèmes multipoints, puis évaluer leurs avantages et inconvénients.

1.3.2 Les technologies tactiles

Comme nous allons le voir, il existe trois types de technologies pour obtenir une surface tactile : acoustique, électronique et optique. Nous allons tout d'abord présenter les différentes variantes propres à chaque technologie, puis effectuer une comparaison de leurs forces et de leurs faiblesses.

La technologie acoustique

Deux variantes existent pour la technologie acoustique. Tout d'abord, nous trouvons la variante des "ondes acoustiques de surface" ou *SAW* (*Surface Acoustic Wave*) qui consiste à injecter dans un substrat (généralement du verre) un onde ultrasonique [84]. Lorsqu'un doigt touche la surface, celui-ci absorbe partiellement l'énergie de l'onde. Des transmetteurs et des récepteurs sont présents de chaque côté de manière à pouvoir mesurer cette variation afin d'en déduire la position du contact. Ce type d'équipement est très solide mais seulement capable de détecter un seul contact. La détection d'objet est également impossible.

7. `http://www.gartner.com/DisplayDocument?id=1288713`

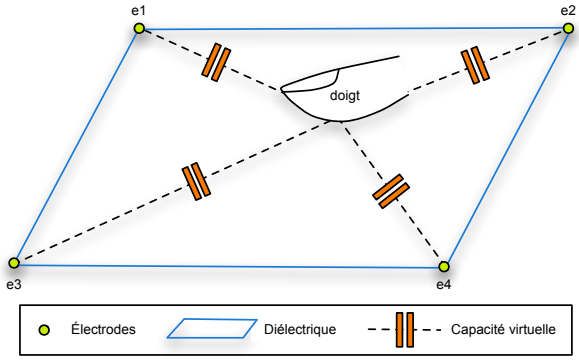


FIGURE 1.10 – Illustration du la technologie tactile capacitive.

La seconde variante est dite à "Reconnaissance d'Impulsions Acoustiques" ou *APR* (*Acoustic Pulse Recognition*) [8]. Un dispositif *APR* comprend une dalle en substrat rigide transparent (généralement du verre), avec quatre transducteurs piézo-électriques montés à l'arrière (figure 1.9a). Lorsqu'un contact se fait sur l'écran, la friction du doigt (ici, possibilité d'utiliser également un stylet) génère une onde acoustique qui est captée par les transducteurs. La position du contact est ensuite déterminée en comparant l'onde reçue avec des échantillons pré-enregistrés. Cette technologie est plutôt imprécise, dépend grandement du type de substrat employé et est très dépendante du bruit environnant.

La technologie électronique

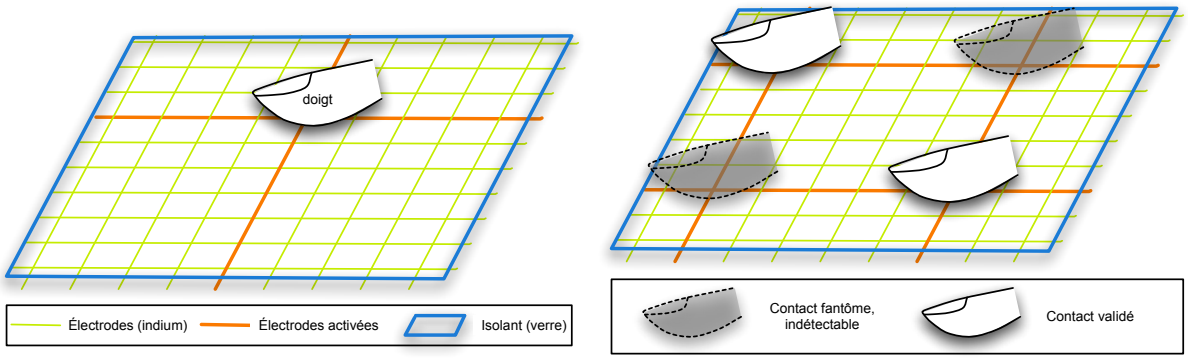
Les technologies électroniques sont les plus utilisés pour les produits commerciaux. Nous allons présentés ses deux variantes principales.

La première est appelée technologie "résistive" et fait partie des technologies les plus abordables pour un produit commercial. Nous la retrouvons donc sur de nombreux téléphones ou PDAs bon marché. Un écran tactile résistif est constitué de deux surfaces transparentes et conductrices séparées par de minuscules espaceurs responsables de l'isolation entre les deux couches (figure 1.9b). Lorsque l'utilisateur pose son doigt sur la surface, la pression exercée crée un contact entre les deux surfaces. La variation dans les champs électriques de ces deux faces conductrices permet de déterminer les coordonnées du point de contact. Cette technologie est facile à endommager, et ne permet de détecter qu'un nombre limité de points en fonction de sa fabrication [9].

La deuxième technologie est nommée technologie "capacitive" (figure 1.10). Dans les systèmes capacitifs, un diélectrique transparent recouvre la surface. Lorsque l'utilisateur touche la surface, certaines charges du diélectrique sont transférées au doigt. En quittant la plaque capacitive, ces charges créent un déficit quantifiable qui permet de détecter la position du contact à l'aide d'électrodes situées aux coins de la plaque. Les objets conducteurs peuvent également être détectés avec cette technologie.

Plus particulièrement, les produits commerciaux utilisent une autre version de la technologie capacitive appelée "projection de capacité" (*Projected Capacitive*). C'est une tech-

8. `http://www.elotouch.com/Products/Touchscreens/AcousticPulseRecognition/`
9. `http://www.elotouch.com/Technologies/compare_resist.asp`



(a) Technologie tactile *Projected Capacitive*.

(b) Limites de la technologie d'auto-capacité.

FIGURE 1.11 – Illustrations de la technologie tactile capacitive.

nologie qui est chère à produire, car basée principalement sur le métal indium, métal appelé à disparaitre dans les prochaines années [10]. Nous pouvons représenter cette technologie par une grille, où chaque croisement forme une capacité (figure 1.11a). Lors d'un appui, des capteurs présents sur chaque ligne et chaque colonne mesurent la capacité afin de déterminer le point de contact.

Il est important de noter que le choix de capteurs influe énormément sur cette technologie. Ainsi nous observons deux sous-catégories. La technologie dite "auto-capacité" mesure chaque ligne et colonne indépendamment, autorisant ainsi deux contacts au plus (figure 1.11b). Des méthodes permettent néanmoins de contourner ces limitations [8]. L'autre sous-catégorie, appelée "capacité mutuelle", mesure chaque croisement indépendamment et rend possible la détection de plusieurs contacts. Les téléphones et tablettes d'Apple embarquent cette technologie qui est d'une réactivité exemplaire.

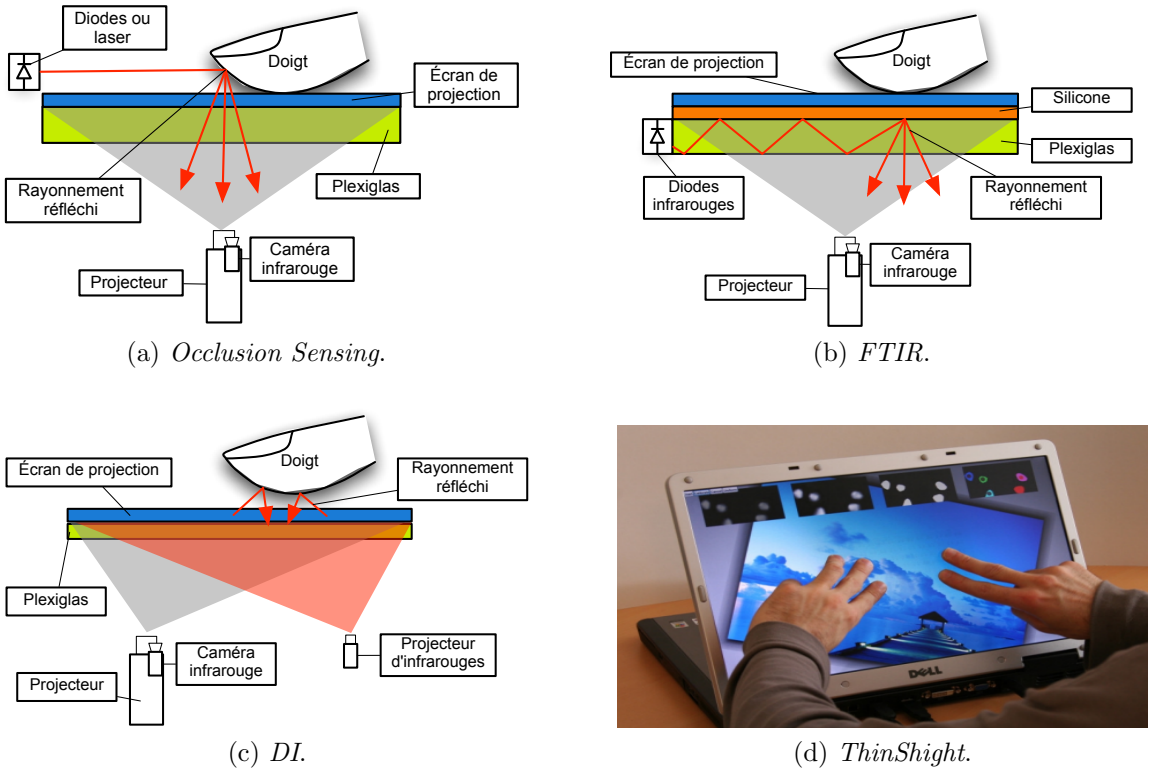
La technologie optique

La dernière technologie est appelée optique du fait du fonctionnement à l'aide de lumière infrarouge et/ou de caméra. Trois variantes existent. Le principe général est similaire pour les différentes variantes : un rayonnement infrarouge est projeté sur une surface, puis sera perturbé par un éventuel contact. Cette perturbation sera ensuite captée par une caméra pour déterminer la position du contact.

La première variante se nomme "détection de l'occultation" (*Occlusion Sensing*). Cette technologie utilise un cadre constitué de plusieurs diodes (ou lasers) qui diffusent de la lumière infrarouge sur la surface de l'écran. Lors d'un appui sur l'écran, la lumière provenant des diodes est arrêtée par le doigt. Ce changement est détecté par un capteur ou une caméra pour ensuite déduire la position du contact (figure 1.12a).

La seconde variante est nommée "réflexion totale" ou *FTIR* (*Frustrated Total Internal Reflection*). Cette variante consiste à émettre un rayonnement infrarouge dans une surface comme du polyméthacrylate de méthyle (*i.e.* plexiglas). Ce rayonnement est enfermé dans la surface (figure 1.12b) et sujet à des déviations causées par d'éventuels contacts sur la

10. `http://tinyurl.com/trouble-touch-screen`



(a) *Occlusion Sensing.* (b) *FTIR.*

(c) *DI.* (d) *ThinShight.*

FIGURE 1.12 – Illustrations des technologies tactiles optiques.

surface. Le rayonnement ainsi dévié est capté par une caméra infrarouge dont l'image sera ensuite analysée à l'aide d'algorithmes de traitement d'image pour déterminer la position des contacts. Cette méthode fut présentée par Jeff Han [42] et rendue populaire par sa facilité de construction et de son faible coût. La détection de plusieurs contacts est possible mais la détection d'objets reste impossible.

La dernière variante est nommée "illumination diffuse" ou *DI* (*Diffused Illumation*). Très proche de la variante *FTIR*, ce système nécessite également un traitement d'image. Des projecteurs infrarouges vont "*illuminer*" la surface. Les contacts utilisateurs perturbent cette illumination qui est captée à l'aide de la caméra (figure 1.12c). Ce système possède l'avantage d'être encore plus facile à construire que la variante *FTIR* mais nécessite un traitement d'image plus important, augmentant la latence du système. La table Surface de Microsoft, dans sa version originale, utilise cette technologie.

La nouvelle version de Surface utilise la technologie ThinShight [50] où chaque pixel de l'écran fait également office de capteur infrarouge, rendant possible la détection de contacts mais également d'objets. Cette variante est l'une des technologies les plus prometteuses. Notons enfin qu'il est possible de produire un retour tactile sur les écrans tactiles comme illustré par le STIMTAC développé dans l'équipe MINT [12, 28, 2, 29].

Technologies / Critères	**Acoustique**		**Électronique**		**Optique**		
	SAW	*APR*	*résistive*	*capacitive*	*occultation*	*FTIR*	*DI*
Multipoint	⊘	⊘	✓	✓	✓	✓	✓
Détection d'objets	⊘	⊘	⊘	✓	⊘	⊘	✓
Solidité	✓	✓	⊘	✓	⊘	⊘	⊘
Latence	☆	☆	☆☆	☆☆☆	☆☆	☆☆	☆☆
Précision	⊘	⊘	✓	✓	⊘	⊘	⊘
Coût	☆☆	☆☆	☆☆	☆	☆☆☆	☆☆☆	☆☆☆
Identification utilisateur	⊘	⊘	⊘	✓	⊘	⊘	⊘

TABLE 1.1 – Tableau illustrant les différences entre les technologies tactiles. Le nombre d'étoiles indique la performance selon le critère. Plus le nombre d'étoiles est élevé, plus les performances du critère sont bonnes.

Avantages - Inconvénients

Le tableau 1.1 récapitule les forces et faiblesses associées à chaque technologie.

Les travaux présentés dans ce mémoire ont été réalisés sur un écran multipoint développé par Immersion [11]. Utilisant la méthode optique DI, notre système était capable de détecter les positions de plusieurs contacts. Malgré la possibilité offerte par la technologie DI, l'utilisation d'objets tangibles n'a pas été abordée pour favoriser une interopérabilité avec les différents équipements multipoints.

Comme nous pouvons le constater, la caractéristique commune à toutes ces technologies est de pouvoir détecter un ou plusieurs contacts. L'usage du doigt de l'utilisateur ainsi que la possibilité de manipuler plusieurs contacts simultanément a entraîné de nombreux travaux de recherche, principalement dans le domaine 2-D. En effet, la surface d'entrée étant située elle-même dans un espace à deux dimensions, cela facilite grandement l'utilisation de techniques d'interaction spécifiques aux environnements 2-D. Nous allons donc maintenant présenter des travaux autour de cette problématique d'interaction 2-D pour les écrans multipoints.

1.3.3 Interaction 2-D et écrans multipoints

Lors de l'utilisation d'écrans multipoints, l'utilisateur se sert de son doigt pour interagir avec les données s'affichant à l'écran. Cela génère un premier problème, celui de l'occultation générée par le doigt : la surface située sur le doigt n'est plus visible par l'uti-

11. `http://www.immersion.fr`



(a) *TapTap* (b) *MagStick*

FIGURE 1.13 – Illustration des techniques *TapTap* et *MagStick* [90].

lisateur [106]. Ce problème, appelé problème du "gros doigt" (*fat finger* dans sa version originale) sera le premier que nous aborderons dans cette section. Ensuite, nous traiterons des techniques de manipulation 2-D, qui nous servirons à établir les idées directrices pour le processus de création de techniques de manipulation 3-D.

Précision

Les écrans multipoints souffrent d'un défaut inhérent à leur nature. En effet, lorsque l'utilisateur pose un doigt sur la surface tactile, celui-ci, de part sa taille, masque une partie de l'écran. Plusieurs solutions pour s'affranchir de ce défaut ont été proposées. Commençons par les solutions spécifiques aux écrans de petites tailles, comme les téléphones.

La solution aujourd'hui utilisée sur les téléphones portables a été introduite par Vogel et al. [104] en 2007. Cette technique, intitulée *Shift*, consiste à afficher l'endroit masqué par le doigt dans une zone où il devient alors visible. Un an plus tard, Roudaut et al. [90] présentent *TapTap* et *MagStick*, deux techniques dédiées à la sélection d'objets sur terminaux mobiles. La première, *TapTap*, consiste en deux temps. Tout d'abord l'utilisateur sélectionne une zone d'intérêt contenant l'objet à sélectionner. Cette zone est ensuite agrandie et centrée au milieu de l'écran pour permettre ensuite la sélection de l'objet désiré (figure 1.13a). La seconde technique, *MagStick*, fonctionne comme une perche télescopique. Cette perche contrôle un pointeur magnétique qui s'accroche sur les objets d'intérêts (figure 1.13b). L'évaluation de ces techniques avec la technique *Shift* révéla leurs supériorités. Pour chacune de ces techniques, la solution du problème de *fat-finger* consiste à passer d'une interaction directe à indirecte. Dans un premier temps l'utilisateur touche une partie de l'écran puis passe en contrôle indirect pour atteindre la zone masquée.

Concernant l'utilisation d'écrans multipoints de plus grandes tailles, plusieurs stratégies ont été présentées [1, 7, 83]. Les derniers travaux dans le domaine abordent des approches similaires. Olwal et al. [83] proposent une technique en deux temps (figure 1.14). Tout d'abord l'utilisateur effectue des va-et-vient (*i.e.* frottement, expliquant le terme *Rubbing* de la technique) pour contrôler le facteur d'échelle de la scène. Une fois qu'une taille suffisante est atteinte, l'utilisateur procède à la sélection de la cible voulue (*i.e. Tapping*).

Malacria et al. [68] ont conceptualisé ces mouvements avec la famille d'interaction *Cyclo**. Les techniques de cette famille utilisent les propriétés d'un geste oscillatoire (*i.e.* fréquence) pour contrôler les variables d'interaction. Deux variantes ont été présentées



FIGURE 1.14 – Illustration de la technique *Rubbing and Tapping* [83].

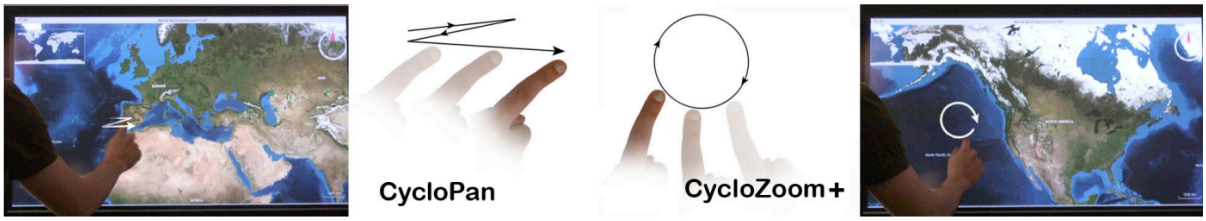


FIGURE 1.15 – Illustration des techniques de la famille *Cyclo** [68] : *CycloPan* et *CycloZoom+*.

(figure 1.15). La première permet d'effectuer de grands déplacements à l'aide de va-et-vient (CycloPan). L'autre ajoute au déplacement la gestion du facteur d'échelle à l'aide de mouvements circulaires (CycloZoom+) .

Il est important de noter que les derniers travaux dans le domaine utilisent tous des mouvements répétés de faibles amplitudes. Ces mouvements ne nécessitent que peu d'effort de la part de l'utilisateur qui peut effectuer ainsi rapidement un enchaînement. Passons maintenant aux techniques de manipulation 2-D sur écrans multipoints.

Manipulation 2-D

Le visionneuse d'image est aujourd'hui la première application livrée avec des écrans multipoints. Si vous tapez "multipoint" dans Youtube (*i.e.* "multitouch") vous tomberez sur des dizaines de vidéos présentant ce mécanisme d'interaction, devenu aujourd'hui courant (figure 1.16). Ce geste, alliant translation, rotation et zoom, sera nommé par la suite *RST* (*i.e.* Rotate-Scale-Translation). Ce mécanisme interaction est la combinaison des travaux de Krueger [59, 60, 61, 62] (pour le geste de zoom) et de Rekimoto [89].

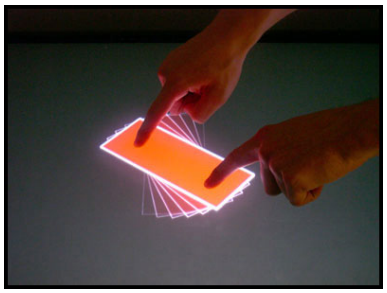


FIGURE 1.16 – Illustration du geste *RST* : Rotate-Scale-Translation.

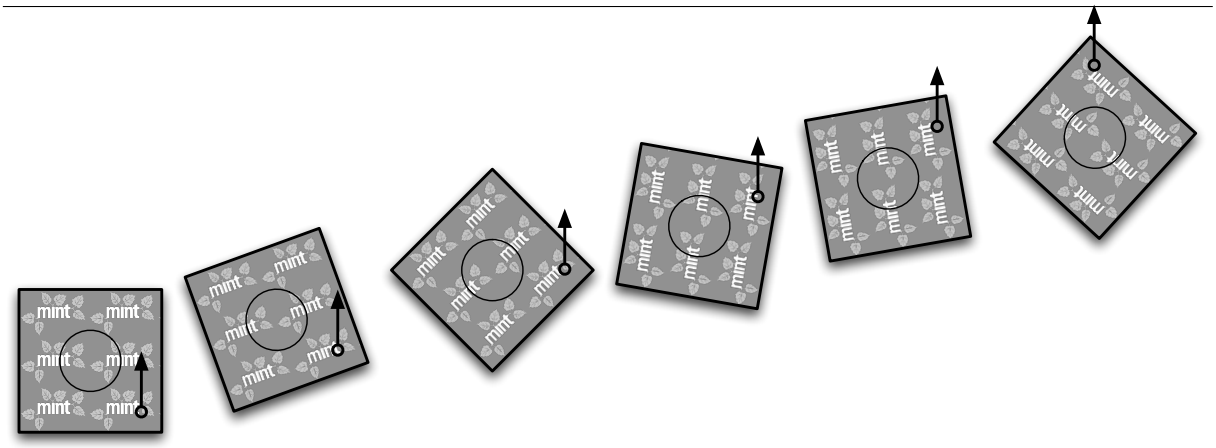


FIGURE 1.17 – Illustration de la technique *RNT* [63].

Plus récemment Kruger et al. [63] ont présenté un mécanisme, intitulé *RNT*, permettant de contrôler à la fois la translation et la rotation avec un doigt unique (figure 1.17). La séparation de contrôle s'effectue en fonction de l'endroit où l'utilisateur pose son doigt. En sélectionnant un objet 2-D en son centre, l'utilisateur translate l'objet. En sélectionnant un objet loin de son centre, celui-ci effectue un mouvement de rotation suivi d'une translation similaire à ce que nous pouvons observer dans la vie réelle. Il s'agit du mécanisme d'interaction programmé par défaut pour les rectangles 2-D sur les tables Microsoft Surface. Nous observons ici une séparation des degrés de liberté de translation de ceux de rotation, fonction de l'endroit manipulé par l'utilisateur. Ce concept est similaire et satisfait les recommandations émises plus tard par Nacenta et al. [79].

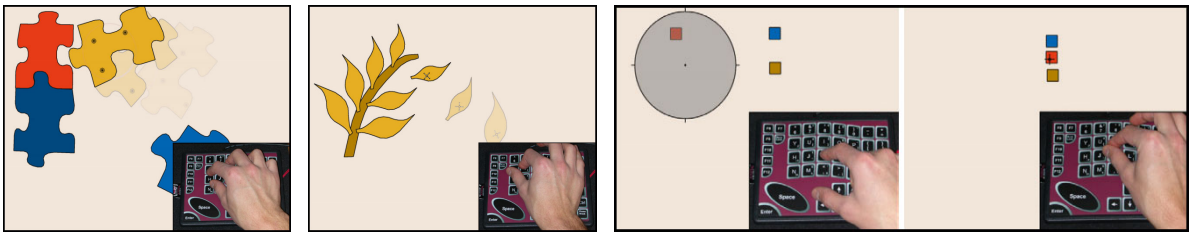
Par la suite, Hancock et al. [45] ont comparé cinq techniques de manipulation 2-D sur écrans multipoints afin de présenter un guide à l'intention des créateurs d'interface 2-D sur écrans multipoints. Leur préconisation consiste à adapter le nombre de degrés de liberté contrôlés en fonction de la tâche à effectuer. Pour les tâches de collaboration ils préconisent ainsi l'usage de la technique *RNT* alors que pour des actions précises, une approche plus séparée semble plus adaptée.

Plus tard, Liu et al. [66] présenteront *TNT*, une technique spécialisée pour la collaboration. Cette technique aborde une stratégie d'intégration des degrés de liberté. Basée sur la reproduction du fonctionnement de la main, cette technique fut démontrée plus rapide que *RNT*.

Il est intéressant de voir que les stratégies varient suivant les approches. Des techniques vont avoir tendance à séparer les degrés de liberté (*e.g. RNT*) tandis que d'autres techniques visent avant tout l'intégration (*e.g. TNT*).

Dans un registre légèrement différent, les travaux de Moscovich [78] se sont intéressés à la manipulation 2-D avec l'utilisation d'un pavé tactile multipoint indirect. Moscovich propose trois nouveaux types de curseurs mettant à profit les multiples contacts possibles sur un tel périphérique.

Le premier se nomme "curseur de main" (*Hand Cursor*) et affiche un ensemble de points sur l'écran, points qui correspondent aux contacts sur le pavé tactile. Ces points permettent à l'utilisateur de contrôler les éléments graphiques présents à l'écran comme il le ferait dans le monde réel. La figure 1.18a montre un utilisateur manipulant des pièces



(a) *Hand Cursor* (b) *Similarity Cursor* (c) *Adjustable Area Cursor*

FIGURE 1.18 – Les différents curseurs introduits par Moscovich [78].

virtuelles de puzzle. La manipulation est effectuée comme si celui-ci manipulait une pièce réelle de puzzle sur une table. Les points multiples peuvent également servir à manipuler plusieurs objets ou à modifier plusieurs paramètres. La distinction des points de contact des deux mains permet d'effectuer des actions simultanées.

Le second curseur introduit est appelé "curseur de similarité" (*Similarity Cursor*). Ce curseur est conçu pour que l'utilisateur se concentre sur la manipulation d'un seul objet, en offrant la possibilité de manipuler simultanément la position, la rotation et le facteur d'échelle (figure 1.18b). Ce curseur n'est pas sans rappeler le geste *RST* à la différence prêt que l'interaction est ici indirecte et nécessite donc de pouvoir contrôler la position du curseur au lieu de toucher directement l'objet souhaité.

Le dernier curseur est nommé "curseur à surface ajustable" (*Adjustable Area Cursor*). Il est conçu pour pouvoir interagir avec plusieurs objets indépendamment de leur taille. L'écartement entre les différents points de contact règle la surface d'action du curseur ce qui permet à la fois de pouvoir sélectionner de petits objets mais également plusieurs objets simultanément. Contrairement aux deux précédents curseurs qui étaient dédiés à la manipulation 2-D d'un objet, ce curseur est lui plus approprié à la sélection.

Ces curseurs, ainsi que les techniques de manipulation 2-D précédemment présentées, nous montrent quelques applications qui sont nées avec la maturation des périphériques multipoints. Les multiples contacts possibles, que ce soit avec une main ou deux, sont mis à profit afin de paralléliser des actions jusqu'alors effectuées de manière séquentielle. L'augmentation de la bande passante d'entrée est utilisée dans ce contexte pour contrôler un plus grand nombre de paramètres responsables de l'interaction. D'une certaine manière, les contacts d'entrées sont reliés aux attributs des objets contrôlés. Pour la manipulation 2-D, il est important de noter que les attributs des objets se trouvent dans un plan 2-D, tout comme les contacts d'entrée. Le passage à la troisième dimension pose alors un nouveau problème, celui de la gestion de la profondeur. Voyons maintenant comment a été adressée cette problématique liée à la manipulation 3-D.

1.4 Techniques de manipulation 3-D multipoints

Nous allons présenter dans cette section les techniques de manipulation 3-D. Dans un premier temps nous nous intéresserons aux techniques offrant une manipulation indirecte, puis nous étudierons les techniques de manipulation directe.

1.4.1 Techniques de manipulation indirecte

Zelesnik et al. [115] ont introduit des techniques utilisant deux curseurs pour positionner, ainsi qu'orienter des objets 3-D. La manipulation n'est cependant pas complète car contrainte sur un plan défini par les curseurs. Suivant le plan défini, deux degrés de liberté de translation sont contrôlés, ainsi qu'un degré de liberté de rotation (*i.e.* autour de la normal du plan). L'utilisateur possède un curseur dans chaque main, qu'il peut contrôler séparément. Il peut ainsi définir un axe de rotation en utilisant le curseur situé dans sa main non-dominante sur un objet, puis faire tourner l'objet autour de cet axe en utilisant le curseur présent dans sa main dominante. Le positionnement est effectué de deux manières différentes. Soit l'utilisateur peut bouger les deux curseurs simultanément, soit il peut utiliser uniquement le curseur de la main non-dominante et relâcher celui de la main dominante. Pour cette technique, il est intéressant de remarquer la séparation de contrôle des attributs de position et d'orientation, qui est présente bien que non discutée.

Plus récemment, Knoedel et al. [58] ont effectué une étude sur l'impact de la manipulation directe sur la réalisation du geste RST. Ils ont comparé deux visualisations (2-D et 3-D) et deux types de manipulation (directe ou indirecte) lors de la réalisation d'une tâche de placement. Les résultats indiquent deux tendances. La manipulation directe améliore le temps de réalisation tandis que la manipulation indirecte améliore la précision ainsi que la trajectoire. Nous observons ici une conséquence intéressante : la trajectoire ne semble pas liée au temps de réalisation dans le sens que nous pourrions naïvement imaginer.

Passons maintenant aux techniques de manipulation directe, utilisant toutes des écrans multipoints.

1.4.2 Techniques de manipulation directe

Hancock et al. [43] ont développé des techniques d'interaction utilisant un, deux et trois doigts. Ces techniques permettent de manipuler un objet 3-D en contrôlant cinq degrés de liberté. Le sixième degré de liberté, Tz, est laissé comme option et son usage n'est pas mentionné dans l'évaluation menée. Avec un doigt, la technique contrôle cinq degrés de liberté en adaptant l'algorithme RNT précédemment exposé [63]. Ces cinq degrés de liberté sont contrôlés intégralement. Avec deux doigts, la technique réutilise également l'algorithme RNT [63] mais d'une manière différente. Le premier doigt contrôle intégralement les degrés de liberté Tx, Ty et Rx. Le second doigt offre un contrôle séparé des deux degrés de liberté de rotation manquants. Enfin, la technique utilisant trois doigts permet de manipuler de manière concurrente position et orientation. Le premier doigt contrôle intégralement les degrés de liberté Tx et Ty. Le deuxième doigt permet le contrôle séparé de Rx et le troisième doigt gère les degrés de liberté Ry et Rz séparément (figure 1.19). Les différentes techniques ont été évaluées dans une expérience contrôlée : la technique utilisant trois doigts est plus rapide, plus précise et préférée parmi les participants.

Afin de pouvoir contrôler les six degrés de liberté nécessaires à une manipulation complète, Hancock et al. introduiront plus tard une technique nommée *Sticky Tools* [44]. Cette technique propose une approche mélangeant intégration et séparation des degrés de liberté. Nous étudierons plus en détail cette technique dans le chapitre 4, consacré à la manipulation 3-D.

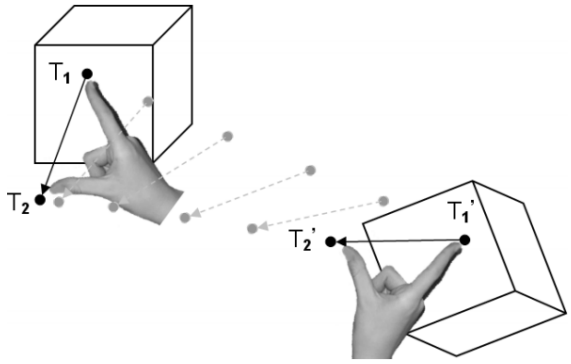
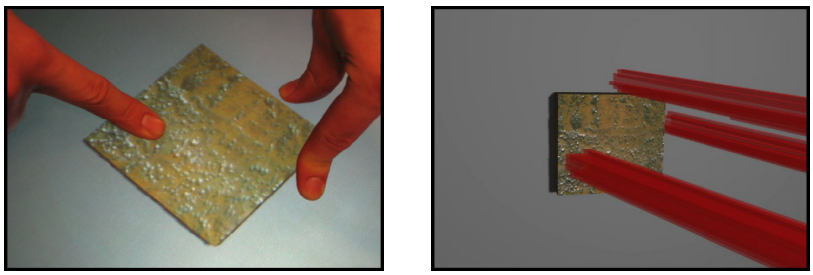


FIGURE 1.19 – Illustration de la technique *Shallow-Depth* [43].

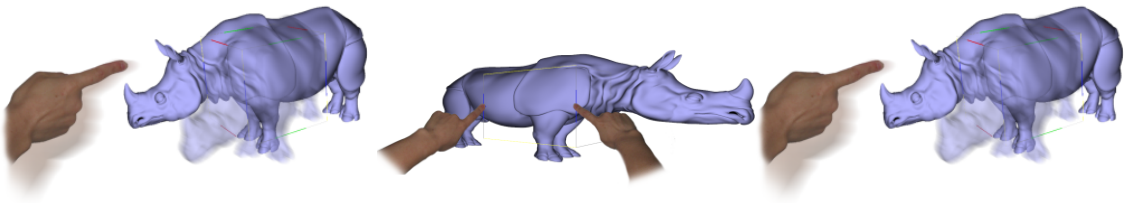


(a) Vue réelle (b) Vue des mandataires

FIGURE 1.20 – Illustration de la technique utilisant un moteur physique [113].

Wilson et al. [112, 113] ont présenté un environnement 3-D capable de réagir avec les doigts des utilisateurs (figure 1.20). La particularité de cet environnement est la présence d'un moteur physique, présent pour augmenter le réalisme des techniques d'interactions. Leur technique est capable de modeler de multiples points de contacts ainsi que différentes formes plus sophistiquées, par exemple la main entière. A l'aide du moteur physique, chaque point devient un mandataire responsable de l'interaction (figure 1.20b). Chaque mandataire créé ajoute des contraintes sur les objets 3-D de la scène, restreignant la manipulation de certains degrés de liberté. Les degrés de liberté non-contraints sont manipulés intégralement. Cette technique permet d'utiliser une vaste palette d'interactions, du doigt à la main entière, en passant par des objets tangibles. Les auteurs ont ainsi montré que l'ajout d'une dynamique similaire au monde réel était possible sur les surfaces multipoints. Cependant, ce réalisme introduit, même si très convainquant, rend la manipulation précise très difficile. La gravité, omniprésente dans l'environnement, diminue ainsi l'étendue d'applications potentielles de leur technique.

Toujours dans un environnement possédant un moteur physique, Hilliges et al. [47] ont présenté une technique pour manipuler les objets en profondeur, malgré la gravité présente. En combinant les reflets émis par les mains sur les tables multipoints utilisant une technologie optique et un traitement d'image, ils sont capables de détecter l'ombre ainsi que la hauteur de la main de l'utilisateur au dessus de la table. En implémentant la détection d'un geste particulier de la main au dessus de la table, la technique permet ainsi de soulever les objets, effectuant une action jusqu'alors impossible. En ajoutant un contrôle par le geste au dessus de la table, cette technique permet un contrôle séparer de



(a) Translation (b) Mise à l'échelle (c) Rotation

FIGURE 1.21 – Illustration de la technique *tBox* [35].

la profondeur (Tz), utilisée en combinaison avec la technique de Wilson. Il s'agit ici plus d'une extension que d'une technique à part entière.

En partant du constat que l'interaction RST était devenue le standard pour la manipulation 2-D, Reisman et al. [88] ont introduit un outil permettant de reproduire le comportement du geste RST mais cette fois-ci en 3-D. L'utilisateur a ainsi l'impression que les points 3-D qu'il manipule restent collés sous ses doigts. La technique utilise un solveur de contraintes qui va minimiser l'erreur entre les positions des doigts et leurs projections. Les degrés de liberté sont ainsi contrôlés de manière intégrale. Nous reviendrons en détail sur cette technique dans le chapitre 4, consacré à la manipulation 3-D.

Fiorella et al. [38] ont présenté une technique spécifique aux périphériques mobiles tactiles. La sélection d'un objet 3-D s'effectue avec un doigt. Ensuite, l'utilisateur a la possibilité de positionner l'objet avec deux doigts, et de contrôler la rotation par une *trackball* virtuelle avec un doigt. Le déplacement de profondeur, ici appelé facteur d'échelle, est contrôlé à l'aide du geste RST. L'approche adoptée ici est la séparation. Les degrés de liberté de rotation sont séparés des degrés de liberté de position mais chacun des sous-ensembles est contrôlé intégralement. La technique est comparée à une technique utilisant des boutons et des widgets, mettant en avant la supériorité de l'interface entièrement tactile.

Les techniques d'interactions qui vont suivre ont été présentées dans des travaux postérieurs à ceux effectués dans ce mémoire. Par soucis d'exactitude, nous les mentionnons ici. Cependant, dans les prochains chapitres, ces techniques ne seront pas mentionnées.

Cohet et al. [35] ont présenté une technique intitulée *tBox* (figure 1.21). Cette technique combine l'utilisation de widgets 3-D avec la manipulation directe. En manipulant la boîte englobante d'un objet, l'utilisateur est ainsi capable de déplacer, d'orienter mais également de mettre à l'échelle l'objet suivant chacune des dimensions de l'espace. La stratégie adoptée ici est la séparation des degrés de liberté. C'est le point de contact qui va déterminer l'axe à utiliser, fonction de la proximité avec l'arrête associée à la boîte englobante.

Dans le cadre d'une application concrète, Kin et al. [57] ont introduit une interface multipoint pour faciliter la création d'environnement utilisé dans les films d'animation. L'interface créée permet de charger un catalogue d'objets qui seront accessibles ensuite pour la création de la scène. A l'aide de contrôles multipoints, l'utilisateur peut déplacer la caméra ainsi que manipuler des objets. La manipulation d'objets s'effectue à deux mains. En utilisant deux doigts en contact direct avec l'objet l'utilisateur peut déplacer l'objet dans le plan de la caméra. Le déplacement en profondeur est permis grâce à l'utilisation

d'un doigt indirect. La rotation est gérée par un doigt direct grâce à la métaphore de la sphère virtuelle. L'usage du geste *RST* permet la mise à l'échelle. La stratégie adoptée ici est la séparation fonction du nombre de doigts en contact avec l'objet et de la présence de doigt indirect. Le choix des différentes métaphores de manipulation est effectué de la manière suivante : les actions les plus fréquemment effectuées sont les moins coûteuses (*i.e.* utilisent moins de doigts).

Les techniques présentées abordent différentes stratégies concernant la gestion des degrés de liberté nécessaires à la manipulation 3-D. Ainsi, la technique *Shallow-Depth* [43] propose une stratégie de séparation des degrés de liberté tandis que la technique de Wilson et al. [113] utilise une stratégie d'intégration. Dans tous ces travaux néanmoins, le choix d'une telle stratégie n'est pas discutée. Dans ce mémoire, nous allons particulièrement nous intéresser à cette problématique. Mais avant de rentrer dans le vif du sujet, examinons les travaux abordant la notion d'intégration et de séparation des degrés de liberté.

1.5 Intégration / séparation des degrés de liberté

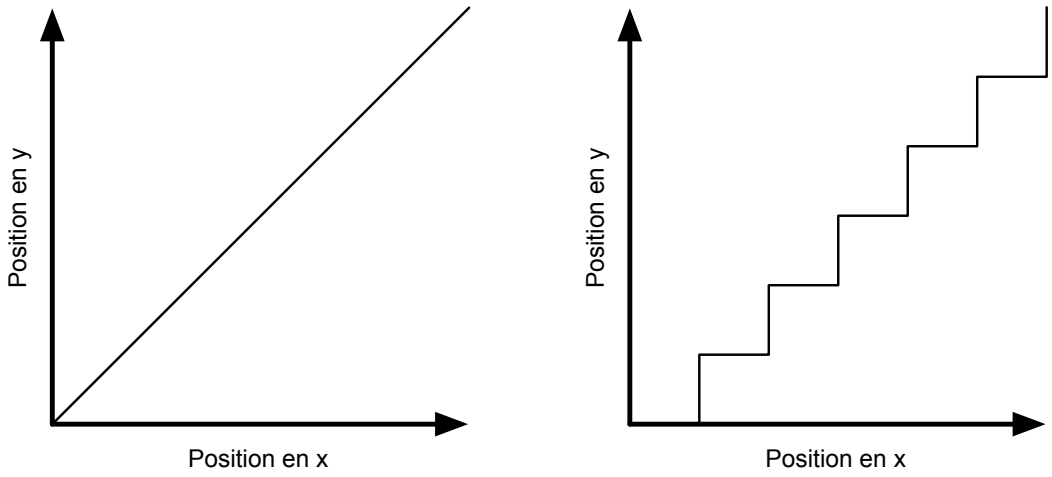
Dans cette section, nous détaillerons dans un premier temps les différentes taxonomies existantes pour caractériser les périphériques d'entrée. Ensuite, nous parlerons de perception utilisateur et exposerons les théories relatives à ce domaine. Enfin, nous présenterons les différents travaux traitant des stratégies de séparation et/ou d'intégration des degrés de liberté, premièrement d'une manière générale, puis spécifique aux écrans multipoints.

1.5.1 Taxonomies

Par soucis de clarté, nous reviendrons sur l'état de l'art spécifique aux taxonomies dans le chapitre 2 dédié à l'introduction d'une taxonomie spécifique aux interactions 3-D pour écrans multipoints. Passons maintenant à l'étude de la perception utilisateur et expliquons les théories associées au domaine.

1.5.2 Structure de perception et structure de contrôle

Un objet multidimensionnel est caractérisé par ses attributs. Un pointeur de souris possède ainsi une taille, une couleur, une forme et une position. En étudiant les travaux sur les structures spatiales de la perception des attributs [3, 91], Garner observa que les attributs de certains objets pouvaient être combinés ensemble, tandis que d'autres attributs restaient distincts et uniques. Il définit ainsi deux structures de perception des attributs. La structure des attributs perçus comme un tout est dite "intégrale" : l'utilisateur perçoit ces attributs de manière globale, intégrale. La structure des attributs distincts et uniques est dite "séparable" : l'utilisateur ne les agrège pas et les perçoit de manière séparée. Par exemple, les positions du pointeur sur les axes x et y sont des attributs perçus de manière intégrale, alors que la forme et la couleur sont perçues séparément. Les attributs intégraux sont en général de natures homogènes tandis que les attributs séparables sont généralement de natures hétérogènes.



(a) Déplacement euclidien : périphérique intégral

(b) Déplacement en marche d'escalier : périphérique séparable

FIGURE 1.22 – Illustration de la notion d'intégrabilité introduite par Jacob et al.

Jacob et al. [54, 55] ont étudié les structures de contrôle des périphériques d'entrée qu'ils définiront d'une manière similaire à Garner. Cette définition s'applique aux périphériques d'entrée possédant au moins deux degrés de liberté. Un périphérique d'entrée possède une structure intégrale s'il est "naturel" (*i.e.* possible) de se déplacer "diagonalement" dans les dimensions de son espace de contrôle. Avec un périphérique intégral, le mouvement effectué par l'utilisateur se situe dans un espace euclidien et modifie toutes les dimensions simultanément (figure 1.22a). Inversement, un périphérique d'entrée possède une structure séparable si le déplacement des dimensions de son espace de contrôle s'effectue en marche d'escalier (figure 1.22b).

Ils évalueront par la suite les différentes combinaisons possibles de structure de contrôle et de perception. Leurs résultats montrent que les périphériques aux structures de contrôle intégrales (réciproquement séparables) sont mieux adaptés aux tâches dont la structure de perception est intégrale (réciproquement séparable). Leur conclusion est la suivante : pour optimiser la réalisation d'une tâche, il faut veiller à l'adéquation de la structure de contrôle avec la structure de perception.

Dans cette sous-section, nous venons d'introduire les notions de structure de perception et de contrôle, d'intégrabilité et de séparabilité. Nous avons également présenté la théorie générale de Jacob reliant périphérique d'entrée et tâche. Passons maintenant à l'étude spécifique des théories liées aux interaction 3-D d'une part, et à l'usage d'écrans multipoints d'autre part.

1.5.3 Théories pour les interactions 3-D et la technologie multipoint

De nombreux travaux ont été effectués sur les périphériques offrant plus de deux degrés de liberté. Ainsi, Zhai et al. [117], Masliah et al. [74], et Wang et al. [107] ont analysé des

tâches de positionnement 3-D. Van Rhijn et al. ont effectué des travaux sur la manipulation d'objets réels pour visualiser des données anatomiques [99]. Mason et al. ont observé les tâches d'orientation d'objets réels [75]. Latulipe et al. ont identifié une amélioration de performance quant à l'utilisation de deux souris pour effectuer une tâche d'alignement à deux mains [64]. Buxton et al. ont examiné les tâches de translation et de mise à l'échelle effectuée à deux mains [19], et Balakrishnan et al. les tâches de suivi à deux mains [5]. Néanmoins, aucun de ces travaux ne s'est intéressé précisément à l'impact de la stratégie d'intégration des degrés de liberté.

Moscovich et Hughes [78], lors de leur étude sur la manipulation multipoint indirecte ont observé que, de part les contraintes physiologiques induites par les mouvements de doigts, il était difficile pour les utilisateurs d'effectuer une rotation à deux doigts tout en gardant une même distance entre les doigts. Le même effet a été remarqué par Ware [108] concernant les manipulations spatiales 3-D. Lors d'une étude utilisateur, Ware a identifié que le positionnement 3-D était plus dur à effectuer lorsque les participants avaient le contrôle de tous les degrés de liberté que lorsque l'un des degrés de liberté était verrouillé.

Basé sur ces remarques, Nacenta et al. [79] créèrent différentes techniques de manipulation de données 2-D sur écran multipoint en adoptant des stratégies de séparation. La première technique effectue la séparation en s'appuyant sur le point de contact avec l'objet, d'une manière similaire à la technique *RNT* [63]. La seconde technique fait appel à un filtre de magnitude. Ce filtre sert à enlever les mouvements involontaires lorsque ceux-ci sont très faibles. La dernière technique est basée sur la reconnaissance de mouvements. L'algorithme est capable de détecter si le geste effectué est davantage un geste de translation, de rotation ou bien encore de mise à l'échelle. Fonction du résultat de l'algorithme, le mécanisme d'interaction adapte son comportement. Lors de l'évaluation, la technique basée sur la reconnaissance de gestes se révéla être le meilleur compromis pour favoriser les performances, le contrôle et la satisfaction utilisateur.

Concernant les interactions 3-D, Teather et Stuerzlinger [96] ont réalisé une étude sur la relation entre le périphérique d'entrée et la tâche de positionnement 3-D, en utilisant à la fois des périphériques d'entrée 2-D et 3-D. Sur cette tâche particulière, ils ont démontré que les périphériques d'entrée à deux dimensions ont le potentiel d'améliorer la vitesse de réalisation de la tâche, si utilisés avec des algorithmes dits "intelligents". Le positionnement 3-D étant une tâche dont la structure peut être considérée comme intégrale [40], ces résultats sont en contradiction avec Jacob. Néanmoins, les algorithmes intelligents mentionnés dépendent de l'objectif à atteindre pour l'utilisateur, difficilement prédictible pour nombre de tâches. Cette technique est également difficile à utiliser dans une scène complexe car les algorithmes peuvent introduire des discontinuités de déplacement du fait du grand nombre d'objets présents.

Veit et al. [100] ont réalisé une étude spécifique à une tâche d'orientation 3-D en comparant deux stratégies pour la manipulation des degrés de liberté. La première stratégie consiste à intégrer tous les degrés de liberté à l'aide d'un périphérique d'entrée à trois dimensions, appliquant à la réalité virtuelle la technique considérée comme la plus efficace [34]. La seconde stratégie adopte une approche de manipulation séparée des degrés de liberté. Les deux stratégies furent évaluées dans une expérience contrôlée. En considérant la théorie de Garner [40], la tâche d'orientation possède une structure intégrale. D'après Jacob [55], la stratégie d'intégration des degrés de liberté devrait donc l'emporter, celle-ci

possédant une structure de contrôle presque identique à la structure de perception de la tâche. Les résultats trouvés vont cependant dans le sens inverse : non seulement les utilisateurs n'arrivent pas à intégrer tous les degrés de liberté avec le périphérique intégral mais en plus la stratégie de séparation améliore le temps de réalisation. Casiez et al. ont par ailleurs montré que les utilisateurs sont capables de coordonner facilement des degrés de liberté séparés [25, 26, 32, 24].

La question concernant la stratégie d'intégration des degrés de liberté pour l'interaction 3-D reste donc en suspend. Nous avons d'un côté la théorie de Jacob qui nous préconise d'intégrer au possible les degrés de liberté de la manipulation 3-D, mais les études réalisées à la fois pour le positionnement 3-D, l'orientation 3-D et la manipulation multipoint nous indiquent que la séparation des degrés de liberté est une alternative viable sinon plus performante. Afin de pouvoir étudier, comparer et quantifier cette séparation, nous allons devoir utiliser différentes métriques. Ainsi la prochaine section sera consacrée à la présentation d'indicateurs servant à mesurer l'intégration des degrés de liberté et plus généralement la coordination.

1.6 Mesurer la coordination

Pour évaluer et comparer les différentes techniques d'interaction 3-D, plusieurs mesures sont traditionnellement utilisées en IHM. Ainsi, le temps d'exécution, le taux d'erreur ou la précision d'exécution constituent des mesures très usitées. Toutefois, si ces mesures permettent de détecter des différences de performances, elles échouent à expliquer l'origine de ces différences. Afin de pouvoir bénéficier d'informations supplémentaires sur l'intégration des degrés de liberté dans la réalisation de la tâche par l'utilisateur plusieurs métriques ont été introduites pour quantifier cette coordination. Autant que possible, nous illustrerons les différentes métriques avec des exemples fictifs.

1.6.1 Intégralité

Jacob et al. [55] ont défini l'intégralité ("*integrality*" dans sa version originale) comme étant la possibilité de modifier plusieurs degrés de liberté simultanément. La figure 1.22 illustre le concept. L'intégralité, comme définie par Jacob, est une mesure indépendante de la tâche. Peu importe que le mouvement rapproche ou éloigne l'objet de son objectif, celui-ci sera pris en compte et sera considéré intégral ou séparable. En ce sens, l'intégralité n'est pas strictement une mesure de coordination mais fut cependant déjà utilisée à cet effet [4].

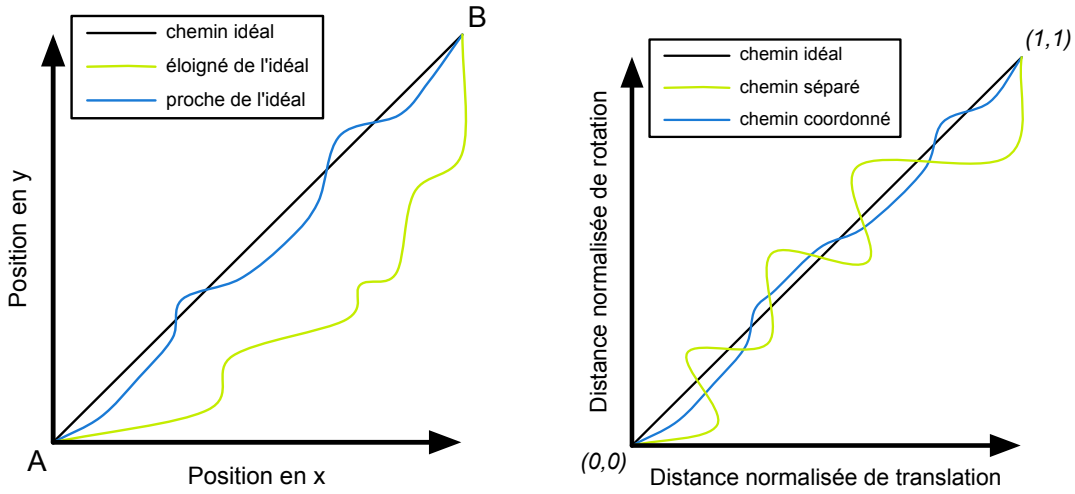
L'intégralité est calculée de la manière suivante. Tout d'abord la trajectoire suivant chacun des degrés de liberté est segmentée à l'aide d'un pas de temps. Ensuite chacun des segments est testé, suivant la présence (ou l'absence) de mouvements supérieurs à un seuil prédéfini. Pour chaque segment de temps, si un mouvement se produit sur tous les degrés de liberté, le segment est classifié "euclidien" (*i.e.* intégral). Le cas échéant, le segment est classifié "marche d'escalier" (*i.e.* séparable). Le résultat final est le ratio du nombre de segments "euclidien", divisé par le nombre de segments "marche d'escalier".

Cette métrique souffre de plusieurs défauts. Tout d'abord elle n'est pas définie lorsque

le nombre de segment "marche d'escalier" est égal à zéro. Ensuite elle ne distingue pas les mouvements qui rapprochent l'objet du but de ceux qui l'éloignent. Enfin, si l'on considère la tâche de manipulation 3-D, elle ne permet pas l'étude d'un sous-ensemble de degrés de liberté. Seul l'ensemble des degrés de liberté est pris en charge dans cette métrique. Afin d'améliorer, la notion d'efficacité a été introduite.

1.6.2 Efficacité

Zhai et al. [116, 117] ont proposé une mesure permettant d'évaluer la coordination lors d'une tâche de placement 3-D : l'efficacité (*efficiency*). Cette métrique dérive de l'affirmation suivante : pour une tâche nécessitant le contrôle de n degrés de liberté, la trajectoire possédant la plus petite distance dans l'espace à n dimensions est considérée comme la plus coordonnée. Pour illustrer ce concept, prenons une tâche de positionnement 2-D, illustrée dans la figure 1.23a. Sur cette figure, la trajectoire optimale est la droite reliant les points A à B. Nous avons représenté deux trajectoires différentes. L'une d'elle est plus proche de la droite idéale, nous dirons donc que son efficacité est plus grande.



(a) Efficacité

(b) Coefficient de Translation/Rotation

FIGURE 1.23 – Illustrations fictives des métriques de Zhai et al.

Basée sur cette observation, tout écart de la trajectoire optimale est considéré alors comme le résultat d'un mouvement qui n'est ni efficace, ni coordonné. Zhai et al. définisse ainsi un coefficient de coordination, représentant l'efficacité du mouvement. De par la nature différente des degrés de liberté de position et d'orientation, deux coefficients sont définis pour une tâche de manipulation 3-D : le coefficient de translation (C_{Tr}) et le coefficient de rotation (C_{Rot}). Ils sont calculés de la manière suivante :

$$C_{Tr} = \frac{\text{Longueur du chemin le plus court}}{\text{Longueur du chemin réel}}$$

$$C_{Rot} = \frac{\text{Amplitude du décalage initial en rotation}}{\text{Amplitude totale des rotations effectuées}}$$

Basée sur cette formulation, une valeur proche de 1 correspond à un mouvement très coordonné, proche de la trajectoire optimale. Respectivement, une valeur proche de 0 indiquera un mouvement très peu coordonné, très éloigné de la trajectoire optimale.

En plus de ces coefficients de coordination, Zhai et al. proposent également un autre indicateur, appelé inefficacité (*'inefficiency'* dans sa version originale). Cette métrique représente le volume de déplacements supplémentaires effectués, c'est à dire l'inverse de la mesure de coordination. Tout comme les coefficients, deux indicateurs d'inefficacité sont présentés, celui de la translation (I_{Tr}) et de la rotation (I_{Rot}). Ils sont définis ainsi :

$$I_{Tr} = \frac{\text{Longueur du chemin réel} - \text{Longueur du chemin le plus court}}{\text{Longueur du chemin le plus court}}$$

$$I_{Rot} = \frac{\text{Amplitude totale des rotations effectuées} - \text{Amplitude du décalage initial en rotation}}{\text{Amplitude totale des rotations effectuées}}$$

Au final, ces indicateurs sont simplement dérivés des coefficients de coordination et n'apportent pas de valeur ajoutée. Voici les relations qui relient indicateurs d'inefficacité et coefficients de coordination.

$$I_{Tr} = \frac{1}{C_{Tr}} - 1$$

$$I_{Rot} = 1 - C_{Rot}$$

Par soucis de simplicité et de clarté, nous utiliserons donc uniquement les coefficients de coordination dans nos prochaines études. Ces deux coefficients de coordination nous révèlent les coordinations respectives des mouvements en translation et en rotation. Cependant, un aspect manque à ces métriques. En effet, un mouvement peut très bien être parfaitement coordonné en translation et parfaitement coordonné en rotation sans que les deux actions ne soient effectuées simultanément. Afin d'obtenir des informations sur ce parallélisme, Zhai et al. [116, 117] ont introduit un troisième indicateur, le coefficient de translation-rotation.

1.6.3 Coefficient de Translation/Rotation

Le coefficient de translation-rotation est défini dans l'espace translation-rotation. Cet espace possède deux dimensions. La première est la distance de translation entre l'objet manipulé et le but à atteindre. La seconde est l'amplitude de rotation entre l'objet manipulé et le but à atteindre. Ces deux dimensions sont normalisées de sorte que les coordonnées $(1, 1)$ représentent le coefficient au début de la manipulation (*i.e.* l'objet est à une distance de translation et amplitude de rotation maximum par rapport à l'objectif) et que les coordonnées $(0, 0)$ représentent l'objet en fin de manipulation (*i.e.* l'objet est correctement positionné et orienté, le décalage entre les deux est égal à 0). Plus formellement, le coefficient de translation-rotation est défini ainsi.

$$C_{Rot}^{Tr} = \frac{\text{Chemin idéal dans l'espace Translation/Rotation}}{\text{Chemin parcouru dans l'espace Translation/Rotation}}$$

Par soucis de clarté, nous avons illustré un exemple dans la figure 1.23b. Une nouvelle fois, le coefficient idéal est représenté par une droite partant des coordonnées $(1, 1)$ aux coordonnées $(0, 0)$. Nous avons tracé sur cette figure deux courbes, représentant deux types de coordination. La première représente un mouvement plutôt coordonné dans l'espace translation-rotation et la seconde illustre un mouvement plutôt séparé.

Le coefficient de translation-rotation présente ainsi une vue d'ensemble de la manipulation des degrés de liberté de translation et d'orientation mais échoue à quantifier précisément la coordination des degrés de liberté. Pour cela, Masliah et al. ont introduit la $\boldsymbol{m}$-metric [74].

1.6.4 *m-metric*

La $\boldsymbol{m}$-metric permet l'étude de sous-ensembles de degrés de liberté. Cette métrique distingue deux aspects durant une tâche de manipulation :

1. La mesure de simultanéité (*'simultaneity of control'* dans sa version originale), qui quantifie le niveau de parallélisme dans la manipulation des degrés de liberté du sous-ensemble étudié.
2. La mesure d'efficacité (*'efficiency of control'* dans sa version originale), qui quantifie le mouvement produit par rapport au chemin optimal.

Masliah et al. définissent la $\boldsymbol{m}$-metric, appelée allocation de contrôle, comme étant le produit des valeurs de simultanéité et d'efficacité. Dans cette définition, la $\boldsymbol{m}$-metric se trouve être une pondération du coefficient de translation-rotation par la mesure de simultanéité. Par comparaison avec le coefficient de translation-rotation, cette métrique nous permettra d'avoir une information sur l'efficacité réelle de la tâche. En effet, seuls les mouvements rapprochant l'objet de son but sont pris en compte. Cette métrique est également normalisée. Une valeur proche de 1 (respectivement 0) indiquera une coordination très efficace (respectivement très peu efficace).

La $\boldsymbol{m}$-metric nous fournit donc un indicateur permettant de quantifier globalement la coordination de l'utilisateur pour réaliser la tâche. Mais la question du nombre de degrés de liberté intégrés ou séparés reste encore en suspend. Pour cela, Veit et al. ont introduits deux métriques.

1.6.5 Nombre de degrés de liberté combinés

Veit et al. [102] ont introduit une métrique permettant d'évaluer le Nombre de Degrés de libertés Combinés (notée *NDC*). Cette métrique est dérivée du calcul de la *MDS* [100] ('Magnitude of Degrees of freedom's Separation' dans sa version originale). Nous allons dans un premier temps détailler cette métrique pour ensuite revenir à la *NDC*.

L'objectif de la *MDS* est de fournir une information quantitative, à chaque instant de la tâche, sur le niveau de séparation des degrés de liberté. Lorsque l'utilisateur manipule un seul degré de liberté, la *MDS* à une valeur élevée. A contrario, lorsque celui-ci manipule simultanément plusieurs degrés de liberté, la valeur de la *MDS* est faible. De part leur nature différente, la *MDS* des degrés de liberté de translation et de rotation est calculée indépendamment. La *MDS* de translation sera extraite à partir des vitesses linéaires des

degrés de liberté associés, tandis que la *MDS* de rotation sera extraite à partir des vitesses angulaires. Pour quantifier l'amplitude de la manipulation d'un degré de liberté, Veit et al. utilisent l'intégrale de la courbe de vitesse associée. Elle est définie de la manière suivante pour les six degrés de liberté de la manipulation 3-D :

$$V_{ddl} = \int_{T}^{T+1} v_{ddl}(t)dt, \forall ddl \in \{Tx, Ty, Tz, Rx, Ry, Rz\}$$

Ensuite, des comparaisons deux à deux sont effectuées pour chacun des degrés de liberté de translation (respectivement de rotation) en calculant la valeur absolue de la différence des intégrales. Cette somme est ensuite normalisée pour obtenir un résultat entre 0 et 1. Plus cette valeur est importante, plus l'utilisateur sépare les degrés de liberté. La *MDS* de translation (notée MDS_{Tr}) et la *MDS* de rotation (notée MDS_{Rot}) sont donc définies ainsi :

$$MDS_{Tr} = \frac{|V_{Tx} - V_{Ty}| + |V_{Tx} - V_{Tz}| + |V_{Ty} - V_{Tz}|}{2 \times V_{tot}} \text{ avec } V_{tot} = V_{Tx} + V_{Ty} + V_{Tz}$$

$$MDS_{Rot} = \frac{|V_{Rx} - V_{Ry}| + |V_{Rx} - V_{Rz}| + |V_{Ry} - V_{Rz}|}{2 \times V_{tot}} \text{ avec } V_{tot} = V_{Rx} + V_{Ry} + V_{Rz}$$

La métrique *NDC* est définie simplement par rapport à la *MDS* afin de permettre une lecture plus facile du nombre de degrés de liberté combinés. Cette métrique nous indiquera explicitement combien de degrés de liberté sont manipulés simultanément. Calculées séparément pour la translation et la rotation, nous sommerons les résultats des *NDC* pour obtenir une indication du nombre total de degrés de liberté manipulés simultanément lors d'une tâche de manipulation, compris donc entre 0 (*i.e.* l'objet n'a été ni déplacé ni orienté) et 6 (*i.e.* tout les degrés de liberté ont été modifiés). Plus formellement, voici la définition de la *NDC* :

$$NDC = NDC_{Tr} + NDC_{Rot}$$

$$\text{avec } NDC_{Tr} = \begin{cases} 1 + 2 \times MDS_{Tr} & , \text{ si } V_{Tx} \neq 0 \text{ ou } V_{Ty} \neq 0 \text{ ou } V_{Tz} \neq 0 \\ 0 & , \text{ sinon} \end{cases}$$

$$\text{et } NDC_{Rot} = \begin{cases} 1 + 2 \times MDS_{Rot} & , \text{ si } V_{Rx} \neq 0 \text{ ou } V_{Ry} \neq 0 \text{ ou } V_{Rz} \neq 0 \\ 0 & , \text{ sinon} \end{cases}$$

1.7 Conclusion

Dans ce chapitre, nous avons introduit les bases nécessaires à nos travaux d'études. Nous avons commencé par présenter les techniques de manipulation 3-D existantes que ce soit pour les interfaces 2-D ou la réalité virtuelle. Nous avons ainsi remarqué que les

méthodes actuelles se basent sur la combinaison de plusieurs techniques d'interaction, profitant des atouts propres à chacune d'entre elles. Cette notion de combinaison nous sera par la suite utile pour la création d'une technique de manipulation 3-D.

Ensuite, nous avons présenté les périphériques multipoints. Nous avons conclu que les écrans multipoints étaient aujourd'hui arrivés à maturité et avons exhibé les nouveaux usages engendrés par cette technologie pour la manipulation 2-D. Ainsi, le geste RST est devenu la base des mécanismes d'interaction 2-D en offrant un contrôle simultané de la position, de l'orientation et du facteur d'échelle.

Puis, nous sommes passé de l'interaction 2-D à l'interaction 3-D, changement qui nécessite de gérer un attribut supplémentaire : la profondeur. Nous avons présenté les techniques existantes en orientant notre démarche autour de la séparation des degrés de liberté. Nous avons conclu en remarquant que les différentes techniques abordaient la manipulation avec des stratégies d'intégration différentes, sans forcément discuter leur choix.

Ce mémoire s'intéressant tout particulièrement à l'intégration des degrés de liberté pour la manipulation 3-D, nous avons ensuite exposé les différents travaux s'intéressant au choix de la stratégie d'intégration. Nous avons ainsi découvert que les derniers travaux soulevaient des contradictions avec la théorie de Jacob, théorie depuis longtemps admise par la communauté scientifique. La question d'intégration des degrés de liberté pour réaliser des tâches de manipulation 3-D sur écrans multipoints se montre donc très pertinente, puisqu'à ce jour nous ne disposons d'aucune information sur ce sujet.

Ce chapitre d'état de l'art s'est terminé sur la présentation de différents indicateurs permettant de qualifier qualitativement et quantitativement la coordination de l'utilisateur lors de la réalisation d'une tâche 3-D. Quantitativement, nous sommes ainsi capables d'évaluer séparément l'efficacité en déplacement et en rotation, le nombre de degrés de liberté combinés et la coordination sur l'ensemble de la tâche. Qualitativement, nous disposons d'un outil qui permet de représenter dans un espace à 2-D la trajectoire en position et en orientation au cours du temps. Ces métriques nous servirons par la suite pour analyser nos nouvelles techniques.

Dans le prochain chapitre, nous allons introduire une nouvelle taxonomie, spécifiquement créée pour représenter les interactions 3-D pour écrans multipoints afin de pouvoir identifier des facteurs clés dans leurs conceptions.

Chapitre 2

Taxonomie des techniques d'interaction 3-D pour écrans multipoints

Sommaire

2.1 Introduction 37
2.2 État de l'art 38
2.3 Notre démarche 40
2.3.1 Constat 40
2.3.2 Proposition d'une taxonomie 42
2.4 Présentation de la taxonomie 43
2.4.1 Gestion des critères 43
2.4.2 Illustration des techniques existantes 44
2.5 Discussion et conclusion 45
2.5.1 Extensions possibles 45
2.5.2 Limites 48
2.5.3 Conclusion 48

2.1 Introduction

Nous allons, dans ce chapitre, proposer une nouvelle taxonomie afin de faciliter la représentation des techniques d'interaction 3-D pour les écrans multipoints et afin d'identifier les composantes clés qui nous seront nécessaires pour concevoir une nouvelle technique de manipulation 3-D.

Dans un premier temps nous allons effectuer l'état de l'art, laissé vacant dans le chapitre précédent par soucis d'intégrité et de continuité dans la logique. Puis, nous expliquerons notre démarche, partant du constat que les taxonomies existantes sont mal adaptées aux possibilités offertes par les écrans multipoints. Ensuite nous présenterons notre taxonomie et nos choix de représentation. Enfin, nous illustrerons la taxonomie en reprenant

les techniques existantes. Nous conclurons par une discussion sur les limites et extensions possibles d'une telle taxonomie.

2.2 État de l'art

Foley et al. [39] classent les périphériques suivant les tâches graphiques qu'ils sont capables de réaliser. Cette définition est néanmoins moins adaptée de nos jours où les ordinateurs sont capables de réaliser énormément de tâches graphiques.

De son côté, Buxton [20] a imaginé une classification propre aux périphériques manuels. Cette classification est basée sur les propriétés physiques des périphériques ainsi que sur le nombre de degrés de liberté qu'ils manipulent. Les périphériques sont repartis suivant six critères : continu/discret (un périphérique est discret s'il possède un nombre réduit d'états, très souvent deux), le moyen d'action, la grandeur mesurée, le nombre de degrés de liberté et enfin le type d'interaction. Dans cette définition, un écran multipoint est un périphérique qui est contrôlé par la main, possédant deux degrés de liberté continus par contact, capable de détecter la position et, suivant la technologie employée, la pression ainsi que d'autres informations comme l'orientation.

Plus récemment, Card et al. [22, 23, 67] ont amélioré la classification en la généralisant à l'ensemble des périphériques continus et discrets. Ils proposent de modéliser la communication homme-machine par l'interaction, dans un langage artificiel, de trois agents : un homme, une machine de communication et une application. Ce langage est défini par un vocabulaire de base, par la suite complexifié par des opérateurs de composition.

Le vocabulaire de base correspond aux mouvements primitifs réalisables. Pour un écran multipoint, il s'agit avant tout de déplacements des doigts (i.e. degrés de liberté linéaires) que nous pouvons mesurer de manière absolue (c'est le cas pour un écran multipoint, périphérique direct) ou relative (c'est le cas pour un pavé tactile multipoint, périphérique indirect). A chaque mouvement primitif, Card et al. proposent d'associer une résolution indiquant le nombre d'états possibles. Pour un bouton de souris, la résolution est de deux (bouton enfoncé ou relâché). Pour un écran multipoint, la résolution dépendra de la technologie employée mais reste très élevée. Nous la qualifierons de continue.

Ces mouvements primitifs sont ensuite mis en relation avec les opérateurs de comparaison, qui décrivent les relations entres les différents degrés de liberté. Dans la taxonomie de Card et al., trois opérateurs de composition sont présents : composition de fusion, composition d'agencement et composition de connexion. La composition de fusion est la combinaison de deux périphériques. Ainsi, un contact sur un écran multipoint (i.e. position en 2-D) est considéré comme la composition de deux "*curseurs*" linéaires possédant chacun un degré de liberté. La composition d'agencement correspond à la présence de plusieurs périphériques sur un même dispositif. Un écran écran multipoint est donc, par définition, l'agencement de plusieurs contacts. Enfin, la composition de connexion apparaît quand le domaine de sortie d'un périphérique correspond au domaine d'entrée d'un autre périphérique. Pour un écran multipoint contrôlant une application de manipulation d'objets 3-D, les contacts sont en connexion avec les six degrés de liberté de la tâche à effectuer.

Un périphérique est représenté par un ensemble de cercles reliés ensemble. Chaque

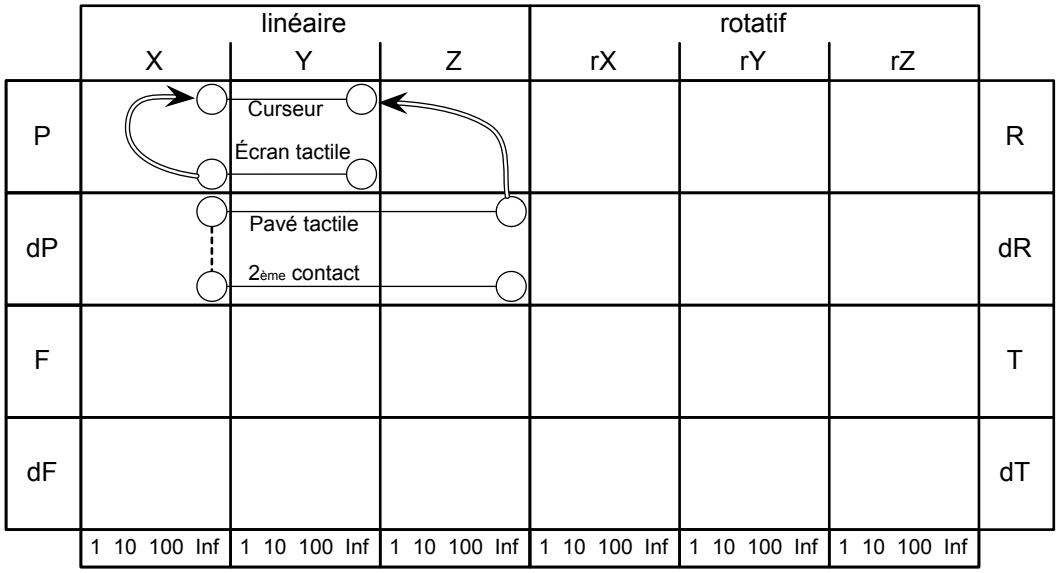


FIGURE 2.1 – Illustration dans la taxomie de Card et l. [23] d'un écran et d'un pavé tactile contrôlant un pointeur de souris à l'écran. Un second contact a été ajouté au pavé tactile pour représenter l'opérateur d'agencement.

cercle représente la grandeur physique mesurée. Chaque ligne représente l'opérateur de composition : ligne continue pour l'opérateur de fusion, ligne discontinue pour l'opérateur d'agencement et double ligne pour l'opérateur de connexion.

Afin d'illustrer cette taxonomie, nous avons choisi de représenter un pavé tactile (multipoint indirect) ainsi qu'un écran multipoint (multipoint direct) contrôlant un pointeur avec un unique doigt (figure 2.1).

Pour le pavé tactile, nous avons donc deux cercles situés dans la ligne dP (i.e. delta de position), celui-ci étant uniquement capable de détecter des mouvements relatifs en position (le pavé tactile fait office ici de souris). Pour l'écran multipoint la représentation est différente. En bénéficiant de la manipulation directe, les mouvements de position sont absolus. Nous retrouvons donc deux cercles mais cette fois-ci dans la ligne P. Pour illustrer l'opérateur d'agencement, nous avons choisi de représenter un second contact pour l'écran multipoint, également en position absolu. Enfin l'opérateur de connexion représente la manière dont les contacts (sortie de périphérique) sont assignés au curseur affiché à l'écran (entrée d'un autre périphérique). Les deux connexions différent suivant les périphérique. Tandis que le contact sur l'écran multipoint se situe dans le même plan spatial ; que nous nommerons pour l'exemple xOy ; le contact de pavé tactile s'effectue, usuellement, dans un plan orthogonal, nommé ici xOz (le placement physique du pavé tactile est généralement orthogonal à l'écran d'affichage).

Passons maintenant à la présentation de la démarche adoptée pour proposer une nouvelle taxonomie.

2.3 Notre démarche

2.3.1 Constat

Avec l'étude des taxonomies existantes nous nous sommes heurtés à plusieurs problèmes liés à la représentation d'interaction 3-D pour écrans multipoints.

Tout d'abord, la taxonomie de Foley et al. [39] basée sur l'usage et la perception des informations présentes à l'écran n'est aujourd'hui plus adaptée. Les tailles d'écrans ont changé, l'espace de travail s'est développé et nous sommes entrés dans l'ère du bureau graphique (GUI) où le paradigme d'interface WIMP (fenêtres, icônes, menus et dispositif de pointage) s'est démocratisé.

Ensuite, la taxonomie de Buxton et al. [20] permet de modéliser seulement les propriétés relatives aux périphériques d'entrée. La relation entre la tâche à effectuer et le périphérique n'est cependant pas discutée, laissant la question des interactions 2-D et 3-D en suspend.

Cette relation est abordée par la taxonomie introduite par Card et al. [22, 23, 67]. Leurs nombreux travaux sur le domaine ont amené les auteurs à produire une taxonomie très généraliste tenant compte du périphérique d'entrée mais également de la tâche à effectuer. Cependant, en tentant de caractériser les techniques d'interaction 3-D pour écrans multipoints existantes, nous nous sommes aperçus que la généricité de cette taxonomie posait problème pour la représentation des techniques spécifiques aux écrans multipoints.

Pour illustrer notre idée, nous avons choisi de représenter la technique d'interaction 2-D RST. Avec cette technique, l'utilisation d'un doigt direct permet de contrôler la position suivant l'axe x et y (Tx et Ty) de manière intégrale. Avec deux doigts directs, l'utilisateur contrôle, en plus de la position 2-D précédente, la rotation autour de la normale du plan xOy (Rz) ainsi que la mise à l'échelle que nous choisirons de représenter par un déplacement en profondeur (Tz).

Représenter l'interaction à un doigt est triviale, il suffit de relier les deux degrés de liberté de position du doigt aux degrés de liberté de position de l'objet à manipuler. Cependant, lorsque nous tentons de représenter l'interaction à deux doigts, les choses se compliquent.

Dans la figure 2.2, nous avons représenté les deux contacts en utilisant l'opérateur d'agencement (i.e. traits pointillés). Mais lorsqu'il faut relier ce contact avec les degrés de liberté contrôlés de l'objet, l'ambiguïté suivante se présente : nous ne pouvons pas savoir si les deux contacts contrôlent intégralement les degrés de liberté ou bien si le contrôle est effectué séparément. Nous avons également représenté une autre variante (figure 2.3) en indiquant les degrés de liberté Tz et Rz séparément. Une nouvelle fois, l'ambiguïté concernant le contrôle intégré ou séparé se présente.

Dans la figure 2.4, nous avons choisi de représenté les deux contacts avec l'opérateur de fusion, de manière à illustrer le caractère intégral du contrôle. Dans cette représentation, l'interaction à deux doigts est facilement exprimable. Néanmoins c'est maintenant l'interaction à un doigt qui devient ambiguë. Difficile de savoir si les degrés de liberté Tx et Ty peuvent être contrôlés par un doigt ou si les deux doigts sont nécessaires.

Partant de ce constat, nous avons décidé de proposer une nouvelle taxonomie, inspirée de Card et al. mais tenant compte de la spécificité liée à l'utilisation d'écrans multipoints.

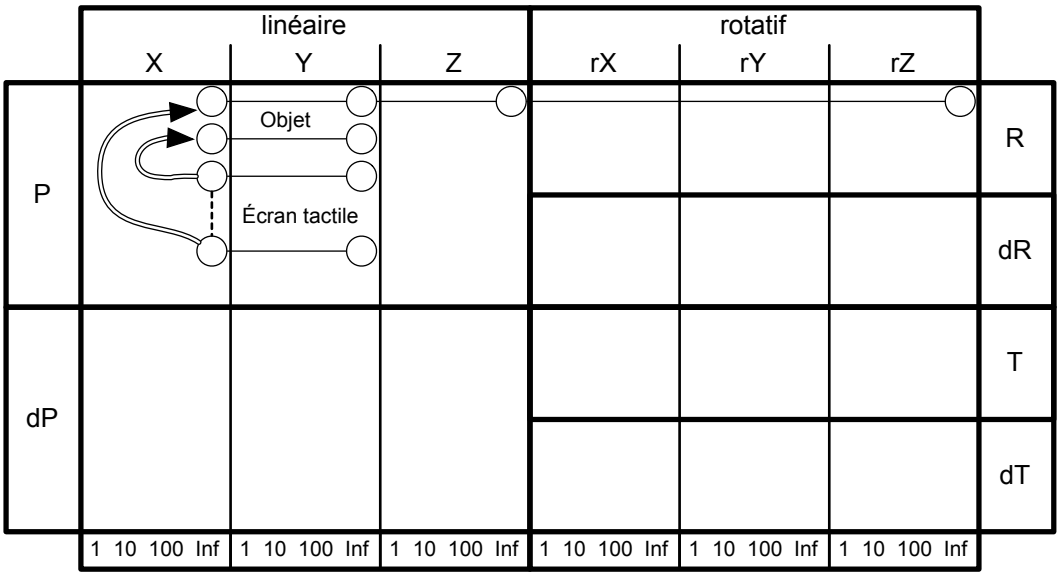


FIGURE 2.2 – Première variante de l'illustration des limites de la taxonomie de Card et al. [23] en représentant le mécanisme d'interaction *RST*. Le deuxième contact est relié ici par l'opérateur d'agencement qui laisse une ambiguïté sur le contrôle des quatre degrés de liberté Tx, Ty, Tz et Rz.

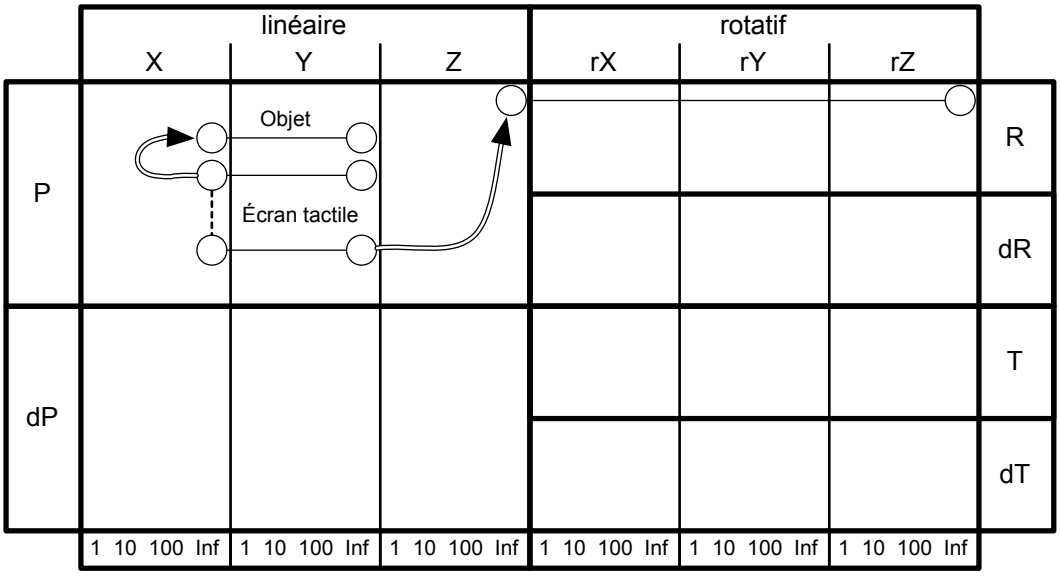


FIGURE 2.3 – Deuxième variante de l'illustration des limites de la taxonomie de Card et al. [23] en représentant le mécanisme d'interaction *RST*. Le deuxième contact est relié ici par l'opérateur d'agencement qui laisse une ambiguïté sur le contrôle des degrés de liberté Tz et Rz.

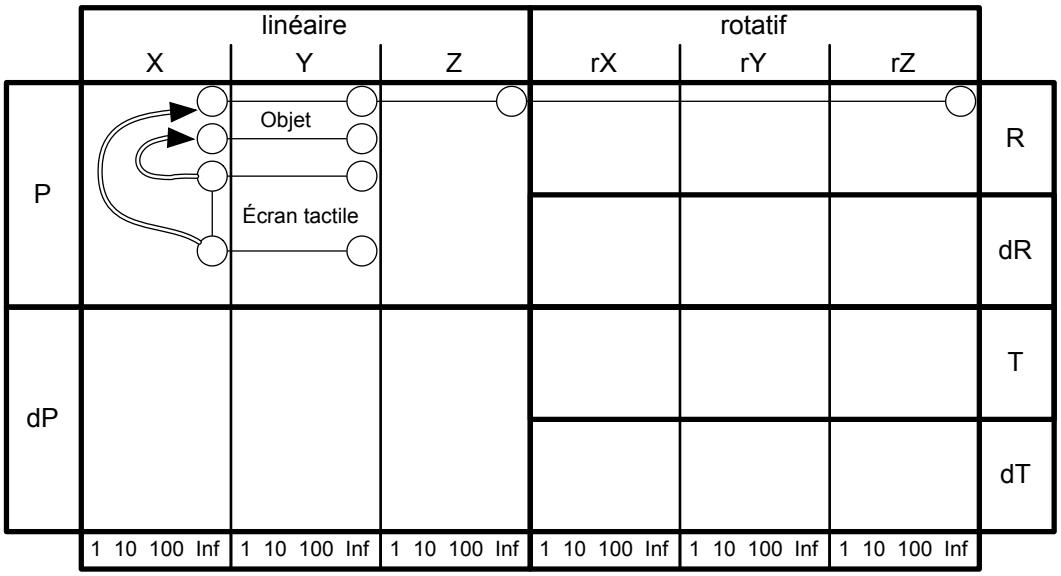


FIGURE 2.4 – Troisième variante de l'illustration des limites de la taxonomie de Card et al. [23] en représentant le mécanisme d'interaction *RST*. Le deuxième contact est relié ici par l'opérateur de fusion qui laisse une ambiguïté sur le contrôle de Tx et Ty.

Le sujet de ce mémoire étant la manipulation 3-D, nous nous intéresserons dans un premier temps à concevoir une taxonomie permettant d'illustrer le contrôle de six degrés de liberté : trois de translation (i.e. Tx, Ty et Tz) et trois de rotation (i.e. Rx, Ry et Rz).

2.3.2 Proposition d'une taxonomie

La taxonomie a été spécialement conçue pour représenter les interactions 3-D spécifiques aux périphériques multipoints directs, c'est à dire les périphériques où l'espace d'affichage et l'espace d'interaction sont confondus. Avant de présenter la taxonomie, nous allons décrire les différents aspects que nous avons identifiés comme nécessaires pour permettre une telle représentation.

Identification du nombre de contacts

Comme illustré précédemment, les taxonomies existantes rendent difficiles la représentation d'interactions multipoints. Le fait que plusieurs contacts soient possibles, simultanément ou séparément, constitue donc le premier critère que notre taxonomie devra gérer.

Directivité

Notre étude des techniques d'interaction 3-D pour écrans multipoints nous a également révélé un autre critère d'importance. Pour l'expliquer, il est important de rappeler le contexte. Nous nous trouvons dans le cadre d'interaction directe, c'est à dire que l'affichage

et la zone d'interaction sont confondus. L'utilisateur peut ainsi indiquer une position à l'écran directement, sans intermédiaire.

Lors de l'utilisation d'écrans multipoints pour de l'interaction 3-D, un autre facteur rentre en jeu. En effet, l'utilisateur peut *directement* toucher l'objet qu'il souhaite manipuler, où alors il peut poser son doigt dans une zone différente afin de contrôler *indirectement* celui-ci. Donc malgré le fait que nous nous trouvons avec un périphérique direct, l'interaction peut, en fonction de la position des contacts, s'effectuer de manière indirecte. Ce critère, que nous appellerons directivité du contact, devra être pris en compte dans la taxonomie.

Manipulation intégrale ou séparable

L'étude de la taxonomie de Card et al. a également mis en avant l'ambiguïté résultant de l'utilisation de plusieurs contacts. En effet, deux fonctionnements sont possibles lors de l'ajout d'un contact sur la surface d'interaction.

Le premier comportement consiste à combiner le nouveau contact avec le contact déjà présent pour former un tout. L'ensemble de ces contacts contrôlera simultanément, *intégralement*, les attributs de l'objet. L'autre comportement consiste à séparer les contacts afin que chacun d'entre eux puisse manipuler des attributs séparément. Notre taxonomie devra intégrer ces aspects d'intégration et de séparation des contacts.

2.4 Présentation de la taxonomie

Dans un premier temps, nous exposerons nos solutions pour la gestion des critères préalablement définis. Ensuite, nous illustrerons cette nouvelle taxonomie en représentant quelques techniques de manipulation 3-D sur écrans multipoints.

2.4.1 Gestion des critères

Identification du nombre de contacts

Nous avons choisi de représenter les techniques en utilisant des modes. Chaque mode représentera un état de la technique, état qui sera défini par le nombre de doigts présents sur la surface. Ce choix est inspiré de Buxton [21] qui représente par des automates les actions possibles de la souris. Similairement, nous pouvons représenter une technique d'interaction par un automate. Cet automate possèdera des modes, qui seront fonction du nombre de contacts présents sur la surface. Notre taxonomie représentera ces modes explicitement.

Directivité

Outre le nombre de contacts présents sur la surface, nous avons identifié que la directivité du contact était un facteur clef. Nous avons donc choisi de représenter chacun des modes fonction du nombre de contacts directs et indirects nécessaires. Ainsi, un mode d'interaction nécessitant deux contacts directs sera nommé mode $2d$, le d indiquant que

les contacts sont directs. De même, un mode d'interaction nécessitant un contact direct et un contact indirect sera nommé mode $1d + 1i$, le i indiquant le contact indirect.

Manipulation intégrale ou séparable

Pour chaque mode d'interaction, nous allons représenté les degrés de liberté manipulés avec de petits cercles, adoptant une notation similaire à Card et al. Si les degrés de liberté sont manipulés intégralement, alors nous allons simplement les reliés ensemble par un trait, de manière similaire à l'opérateur de fusion dans la taxonomie de Card. Si certains degrés de liberté sont manipulés séparément, alors aucun trait ne sera présent pour illustrer leur caractère séparé. Afin de distinguer les contacts indirects, nous marquerons d'un i les cercles correspondant aux degrés de liberté contrôlés par ces contacts.

En combinant les trois critères précédemment décrits, nous sommes ainsi capables de représenter les techniques d'interaction pour les écrans multipoints. Nous allons maintenant fournir quelques exemples pour illustrer concrètement la taxonomie.

2.4.2 Illustration des techniques existantes

Interaction *RST*

Dans un premier temps, nous avons choisi de représenter le mécanisme d'interaction *RST*. Comme présenté à la section 2.3.1, cette représentation était impossible dans la taxonomie de Card. La figure 2.5 représente la technique dans notre taxonomie. Deux modes sont présents. Avec un contact direct (mode $1d$), l'utilisateur peut déplacer un objet dans un plan 2-D, contrôlant intégralement les deux degrés de liberté nécessaires. Lorsque deux contacts sont présents sur la surface (et plus généralement si n contacts sont présents), l'utilisateur contrôle intégralement quatre degrés de liberté (deux de position, plus un de rotation et un de mise à l'échelle, représenté ici par le déplacement en profondeur).

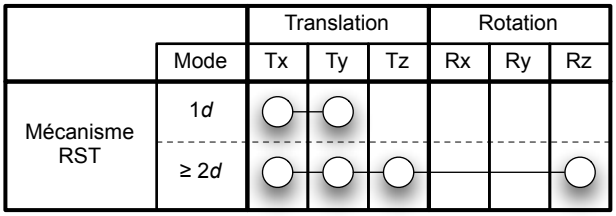


FIGURE 2.5 – Illustration du mécanisme d'interaction *RST* dans notre taxonomie.

Interaction *Shallow-Depth*

Notre deuxième exemple se fera avec la technique introduite par Hancock et al. [43], intitulée *Shallow-Depth*. L'illustration se trouve dans la figure 2.6. Trois modes sont présents pour cette technique. Avec un doigt direct (mode $1d$), l'utilisateur peut déplacer l'objet sur deux dimensions, contrôlés intégralement. Avec l'ajout d'un doigt indirect

(mode $1d + 1i$), l'utilisateur contrôle en plus, toujours intégralement, le degré de liberté associé à la rotation orthogonale au plan de l'écran. L'option de translation en profondeur proposée par Hancock et al. est également représentée. Avec l'ajout d'un deuxième doigt indirect (mode $1d + 2i$), l'utilisateur contrôle séparément les deux degrés de liberté d'orientation restant Rx et Ry.

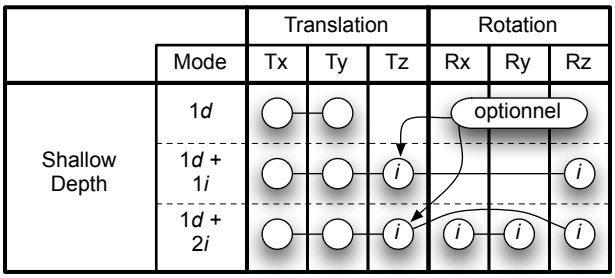


FIGURE 2.6 – Illustration de la technique *Shallow-Depth* [43] dans notre taxonomie.

Interaction avec moteur physique

Notre dernier exemple sera la technique présentée par Wilson et al. [113], illustrée dans la figure 2.7. Dans cette technique, le critère de directivité n'est pas précisé car non pertinent. En effet, les contacts font office de mandataires restreignant la manipulation, qu'ils soient directs ou indirects. La manipulation étant simulée par le moteur physique où une force de gravité est présente, l'utilisateur ne peut donc pas contrôler la direction orthogonale à l'écran (Tz ici). Les cinq autres degrés de liberté sont contrôlés intégralement dans un mode unique (mode n où n représente le nombre de doigts).

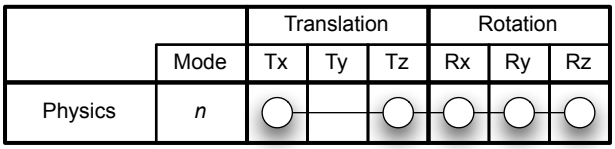


FIGURE 2.7 – Illustration de la technique de Wilson et al. [113] dans notre taxonomie.

2.5 Discussion et conclusion

Nous venons d'introduire une nouvelle taxonomie, dédiée à la représentation de techniques d'interactions 3-D pour écrans multipoints. Nous allons, pour finir ce chapitre, lister quelques extensions possibles afin d'agrandir le périmètre d'action de la taxonomie. Enfin, nous énumérerons les limites de la taxonomie.

2.5.1 Extensions possibles

Pour commencer, nous discuterons d'une extension possible pour intégrer des degrés de liberté supplémentaires. Ensuite, nous nous attarderons à étendre la taxonomie aux péri-

phériques indirects, tels que les pavés multipoints. Enfin, nous discuterons de l'intégration de la manipulation par boîtes englobantes.

Ajout de degrés de liberté

Initialement conçu pour les techniques de manipulation 3-D, nous avions conçu la taxonomie pour gérer seulement six degrés de liberté. Cependant, pour certaines applications de création de scènes 3-D, le facteur de mise à l'échelle est important et représente trois degrés de liberté. Nous proposons donc d'inclure ces degrés de liberté supplémentaires dans notre taxonomie.

Pour les tâches de navigation 3-D, ce ne sont plus des objets qui sont manipulés mais des caméras. Notre taxonomie peut également s'adapter aux tâches de navigation en incluant les six degrés de liberté propres à la caméra. Afin d'illustrer nos propos, nous avons représenter dans la figure 2.8 la forme étendue de la taxonomie pour une technique qui manipulerait à la fois le facteur de mise à l'échelle ainsi que la caméra.

		Translation			Rotation			Mise à l'échelle			Caméra Translation			Caméra Rotation		
	Mode	Tx	Ty	Tz	Rx	Ry	Rz	Sx	Sy	Sz	CTx	CTY	CTz	CRx	CRy	CRz
Technique																

FIGURE 2.8 – Extension de notre taxonomie pour gérer plus de degrés de liberté.

Adaptation aux périphériques indirects

La taxonomie de Card et al. possède la qualité intéressante d'être très générique. La représentation des contacts indirects se fait simplement dans cette taxonomie, les degrés de liberté d'entrée passent de la ligne "position absolue" (P) à la ligne "position relative" (dP).

Nous proposons d'adapter notre taxonomie simplement, en supprimant la notion de directivité des contact pour les périphériques multipoints indirects. En effet, lors de l'utilisation de ce type de périphériques, l'utilisateur ne peut plus toucher *directement* l'objet à manipuler. Pour cependant maintenir l'identification des doigts, nous proposons de marquer les cercles correspondant aux degrés de liberté avec le nombre de doigts associés à leur contrôle. Pour illustrer cette extension, nous avons représenté l'interaction *RST* indirecte, telle que proposée par Knoedel et al. [58], dans la figure 2.9.

Avec l'utilisation d'un doigt (mode 1), l'utilisateur contrôle deux degrés de liberté de position, intégralement. L'ajout d'un second doigt (mode 2) permet le contrôle, toujours intégral, de quatre degrés de liberté. Pour illustrer la notion de séparation, nous avons ajouté un troisième mode, fictif. Dans ce mode (mode 3), le troisième contact contrôle les deux degrés de liberté de rotation manquant, d'une manière similaire à Hancock et al. [43].

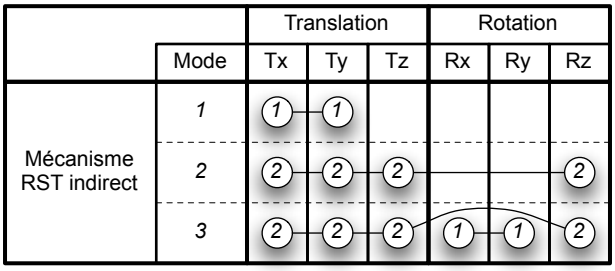


FIGURE 2.9 – Extension de notre taxonomie aux périphériques multipoints indirects : illustration du mécanisme *RST* indirect.

Utilisation de la boîte englobante

Les applications 3-D font régulièrement appelle aux boîtes englobantes pour la manipulation d'objets. Une boîte englobante est représentée par un parallélépipède 3-D qui englobe en son sein le volume entier de l'objet auquel il est rattaché (figure 2.10). Ainsi, les widgets 3-D [36, 95] utilisent la boîte englobante des objets pour effectuer leur manipulation.

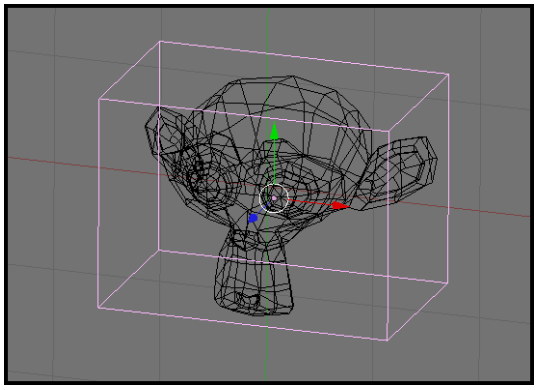


FIGURE 2.10 – Illustration du concept de boîte englobante.

Notre taxonomie, telle que présentée actuellement, ne permet pas de représenter des techniques qui utiliseraient spécifiquement une manipulation par boîtes englobantes. En effet, le critère de directivité se limite à deux possibilités : direct ou indirect. Afin de représenter une éventuelle manipulation de la boîte englobante nous proposons de remplacer le cercle par un carré, forme illustrant le pavé de la boîte englobante.

Ainsi, nous pouvons également représenter des techniques comme *tBox* [35] (figure 2.11). En utilisant un doigt direct sur la boîte englobante (mode $1d$), l'utilisateur contrôle un degré de liberté de position, fonction de l'arrête de la boite sélectionnée. Avec un doigt indirect traversant la boîte (mode $1i$), l'utilisateur manipule un degré de liberté de rotation, déterminé à nouveau par l'arrête sélectionnée. Nous remarquerons l'absence de l'indicateur de directivité, aucune ambiguïté n'étant présente. Enfin, avec deux doigts directs sur la boîte englobante (mode $2d$), la mise à l'échelle suivant un axe peut être effectuée.

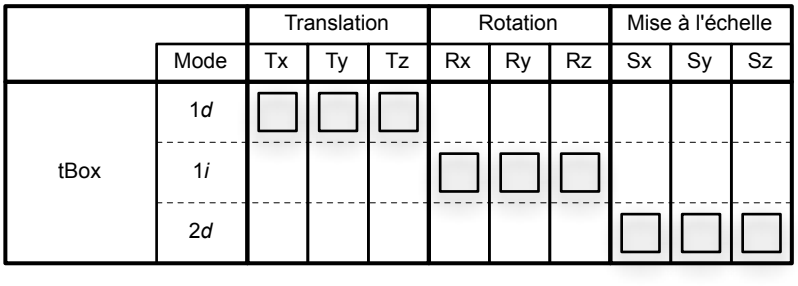


FIGURE 2.11 – Illustration de *tBox* [35] dans notre taxonomie étendue.

2.5.2 Limites

Tandis que la taxonomie de Card et al. trouvait ses limites du fait de sa généricité, notre taxonomie souffre de sa spécificité.

Ainsi, il est impossible de représenter des techniques utilisant des gestes dans la taxonomie, à moins d'introduire de nouvelles notations qui viendraient nuire à la simplicité voulue lors de sa création. Dans la figure 2.7, nous avons représenté la technique de Wilson et al. [113] qui utilise les contacts comme mandataires de la manipulation. L'illustration de cette technique souffre également de la simplicité de représentation. En effet, nous avons choisi de représenter la manipulation des degrés de liberté sans notion de directivité pour les doigts. Cependant, cette approximation entraine une perte d'informations. De manière générale, notre taxonomie n'est pas adaptée aux mécanismes d'interaction utilisant un intermédiaire entre les contacts sur la surface tactile, et les objets manipulés.

2.5.3 Conclusion

Dans ce chapitre nous avons introduit une nouvelle taxonomie afin de représenter les techniques de manipulation 3-D propres aux écrans multipoints. Cette taxonomie a été conçue afin de pouvoir expliciter le fonctionnement des différentes techniques existantes, d'une manière simple et efficace.

Nous avons également introduits des extensions afin de pouvoir prendre en charge des périphériques indirects tel que les pavés multipoints. Nous avons également discuté de l'ajout du contrôle de caméra ainsi que de degrés de liberté supplémentaires pour étendre le champ d'application de la taxonomie. Enfin, nous avons explicité les limites de cette taxonomie, limites intrinsèquement liées à sa création voulue simpliste.

Cet outil, combiné avec les mesures de coordination introduites dans l'état de l'art, va nous permettre de procéder à l'étude des techniques d'interaction 3-D pour écrans multipoints. Le prochain chapitre sera ainsi consacré à une sous-tâche de la manipulation 3-D : le positionnement 3-D. Nous allons étudier les techniques d'interaction offrant le contrôle des degrés de liberté de position : Tx, Ty et Tz.

Chapitre 3

Le positionnement d'objets 3-D

Sommaire

3.1 Introduction 49
3.2 Notre démarche 50
3.2.1 Technique 1 : *MT-viewport* 51
3.2.2 Technique 2 : *Z-technique* 53
3.3 Évaluation des techniques 55
3.3.1 Objectifs 55
3.3.2 Matériel 55
3.3.3 Tâche réalisée 56
3.3.4 Participants 56
3.3.5 Plan expérimental 56
3.4 Résultats et discussion 57
3.4.1 Temps de positionnement 57
3.4.2 Coordination 58
3.4.3 Utilisation de l'extension multipoint de la technique *MT-viewport* 66
3.4.4 Analyse qualitative 67
3.5 Conclusion 68

3.1 Introduction

Le positionnement correspond à une des tâches élémentaires d'une application 3-D. Il permet aux utilisateurs de déplacer des objets dans l'espace de la scène virtuelle en ajustant de manière répétée leur position dans cet espace. Dans les environnements de type bureau, la souris représente le périphérique d'entrée le plus utilisé pour effectuer les actions de positionnement 3-D. La technique la plus utilisée [82] consiste à subdiviser l'écran, présentant plusieurs vues de la même scène 3-D (figure 3.1). Chaque vue, que nous nommerons désormais viewport, permet le contrôle de deux degrés de liberté différents, procurant ainsi le positionnement 3-D en combinant au moins deux vues. Les périphériques d'entrée à trois dimensions offrent une alternative à cette technique et permettent de

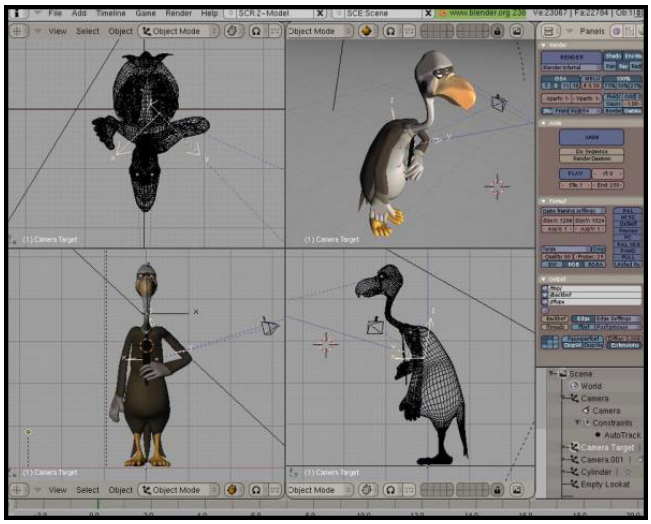


FIGURE 3.1 – Capture d'écran du logiciel Blender illustrant la gamme de technique divisant l'écran en plusieurs vues.

contrôler les 3 degrés de liberté nécessaires de manière intégrale. Néanmoins, ce type de périphériques d'entrée souffre de problèmes de précision, de débrayage et enfin de fatigue.

Jusqu'à récemment, le développement de techniques d'interaction 3-D pour écrans multipoints se focalisait principalement sur la collaboration [66] ou encore la rotation d'objets 3-D [43]. Aucune technique d'interaction pour écrans multipoints n'existait pour réaliser le placement libre d'objets dans un environnement 3-D. Les techniques d'interaction divisant l'écran en différents viewports représentent la gamme la plus utilisée pour les périphériques d'entrée à deux dimensions comme la souris.

Dans ce chapitre, nous allons exposer deux nouvelles techniques d'interaction pour écrans multipoints permettant le positionnement 3-D. Dans un premier temps, nous nous attarderons sur le processus de mise au point de ces techniques. Ensuite, nous évaluerons ces techniques dans une expérience contrôlée. Enfin, nous porterons notre réflexion sur l'intégration et la séparation des degrés de liberté pour la tâche de positionnement 3-D.

3.2 Notre démarche

Notre première étape dans cette étude fût de se focaliser sur la création de techniques d'interactions permettant le positionnement d'un objet 3-D avec un écran multipoint. Durant le début de ce processus, nous avons observé plusieurs utilisateurs manipulant une application d'images sur un écran multipoint. Les conclusions furent sans équivoque : la force d'un tel périphérique réside dans l'interaction directe. L'utilisateur a ainsi la possibilité de manipuler les objets directement en touchant ce qu'il désire. Nous pensons que cet atout offre au grand public une immersion beaucoup plus rapide et amusante, et avons donc développé des techniques d'interactions tirant parti de cette faculté.

La seconde étape a été de rechercher les techniques d'interaction déjà existantes, afin d'identifier celles qui possédaient le potentiel d'être utilisées en interaction directe. Parmi les techniques d'interaction étudiées permettant un positionnement libre, deux s'avérèrent

compatibles avec de l'interaction directe. La première technique a été introduite par Venolia en 1993 [103] (voir l'état de l'art dans section 1.2.1). La seconde regroupe tout une gamme de techniques d'interaction. Il s'agit des techniques mentionnées dans l'introduction, subdivisant l'écran en plusieurs viewports. Ces deux techniques possèdent également une différence intéressante. Elles diffèrent sur la manière de manipuler la profondeur, principale difficulté lorsque l'on travaille avec des données 3-D projetées sur un écran 2-D. Tandis que la gamme de techniques utilisant différents viewports autorise la manipulation directe sur un viewport, la technique de Venolia utilise la manipulation indirecte pour le contrôle de la profondeur. Cela nous a amené à revisiter ces techniques pour les adapter aux écrans multipoints.

Nous allons, dans les sous-sections 3.2.1 et 3.2.2, décrire plus en détail ces techniques. Nous insisterons sur la relation entre périphérique d'entrée et la tâche à effectuer, particulièrement sur la manière dont sont contrôlés les degrés de liberté.

3.2.1 Technique 1 : *MT-viewport*

Beaucoup d'applications dédiées à la conception 3-D (Autocad© [12], Archicad© [13], Blender [14], etc.) proposent une technique d'affichage qui consiste à subdiviser l'écran en trois ou quatre vues (figure 3.1 pour un exemple). Usuellement, les trois premières vues représentent la scène en projection orthographique et sont orthogonales les unes par rapport aux autres. Ce choix provient du savoir historique des professionnels du domaine. En effet, ceux-ci sont habitués à travailler avec des plans 2-D papier, représentant des objets 3-D sous différentes vues : de face, de dessus et enfin de côté. Ces vues sont référencées comme des *viewports*. Chaque viewport permet le contrôle de deux degrés de liberté d'une manière intégrale tandis que le viewport représentant la scène en perspective permet traditionnellement (i.e. mais pas obligatoirement) de contrôler la position dans un plan parallèle au sol. Le viewport perspective permet également de faciliter la visualisation de la scène.

Nous avons choisi d'étendre cette technique afin de tirer profit de l'augmentation de la bande passante d'entrée offerte par les écrans multipoints. La figure 3.2 représente la technique dans la taxonomie précédemment introduite (chapitre 2). Nous allons par la suite détailler chacun des modes.

Viewport traditionnel

Le mode $1d$ représente la gamme de techniques traditionnellement utilisée avec un écran subdivisé en viewports. Nous nommerons ce mode *viewports traditionnels*. Quand un doigt touche la surface, le viewport correspondant est détecté et un rayon orthogonal à la vue est lancé dans la scène. Le premier objet rencontré par le rayon est ensuite sélectionné. Lorsque le doigt effectue un mouvement sur la surface, celui-ci est traduit dans le viewport associé à l'objet. Par conséquent, l'objet est déplacé dans le plan parallèle à la vue passant par le centre de l'objet. Le plan étant différents suivant les vues, cela

12. `http://www.autodesk.fr/adsk/servlet/pc/index?siteID=458335&id=14626647`
13. `http://www.abvent.com/softwares/archicad/`
14. `http://www.blender.org/`

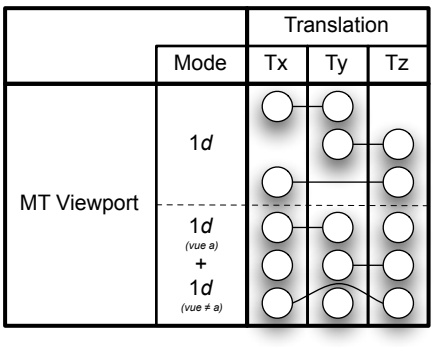


FIGURE 3.2 – Illustration de la technique *MT-viewport* dans notre taxonomie.

explique les différentes combinaisons illustrées dans le mode $1d$ sur la taxonomie (figure 3.2). Le résultat final est identique à une manipulation réalisée à l'aide de la souris, deux degrés de liberté sont contrôlés d'une manière intégrale. L'objet reste sous le doigt tout comme il le ferait avec le pointeur de la souris.

Viewports multipoints

L'extension multipoint de la technique est déclenchée lorsqu'un second doigt vient toucher un autre viewport, représentée par le mode $1d + 1d$ dans la taxonomie. Ce second doigt détermine la valeur du troisième degré de liberté et téléporte l'objet à cette valeur. Le troisième degré de liberté se situant sur un rayon perpendiculaire au viewport du premier doigt, le changement de ce degré de liberté n'affecte pas la position de l'objet dans le premier viewport. L'objet reste donc en permanence sous le premier doigt de l'utilisateur, conservant ainsi la possibilité d'interaction directe avec celui-ci. L'utilisateur peut ensuite ajuster la valeur du troisième degré de liberté en effectuant des mouvements avec le deuxième doigt. Nous avons implémenté un retour visuel sous la forme d'une ligne semi-transparente passant par le centre de l'objet pour indiquer à l'utilisateur les valeurs possibles de ce troisième degré de liberté (figure 3.3).

Dans ce mode, la manipulation simultanée des trois degrés de liberté nécessite donc un mouvement concurrent de deux doigts. Une subtilité doit cependant être détaillée. En effet, si l'utilisateur effectue un mouvement avec le premier doigt, la position de l'objet va être modifiée et, potentiellement, le second doigt ne sera plus en contact direct avec le retour visuel. Afin de toujours permettre une manipulation simultanée, nous ajustons en continu le troisième degré de liberté fonction de la coordonnée manquante au premier doigt (coordonnée représentée par le retour visuel), même si le doigt ne se trouve plus en contact direct avec le retour visuel.

Cette technique utilise donc l'interaction directe pour contrôler tout les degrés de liberté nécessaires au positionnement 3-D de l'objet. Le premier doigt contrôle intégralement deux degrés de liberté tandis que le second contrôle séparément le degré de liberté manquant. Les deux degrés de liberté du premier doigt sont donc intégrés mais le troisième degré de liberté, degré contrôlé indépendamment, ne peut être qualifié selon le paradigme intégré/séparé.

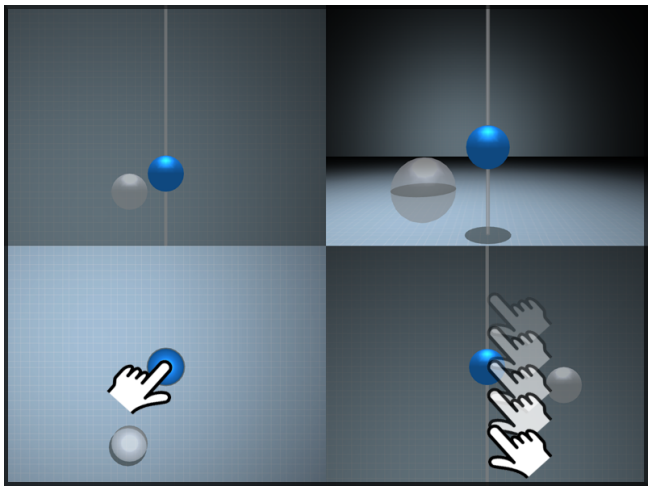


FIGURE 3.3 – Capture d'écran de la technique *MT-viewport*. Le doigt de gauche est utilisé pour un positionnement 2-D dans la plan de caméra correspondant au viewport associé. Le doigt de droite est utilisé pour téléporter le doigt et contrôler le troisième degré de liberté. La ligne grise représente le retour visuel indiquant à l'utilisateur le mouvement autorisé par le deuxième doigt.

3.2.2 Technique 2 : *Z-technique*

Nous avons conçu cette deuxième technique afin de permettre un positionnement 3-D dans une unique vue de la scène, maximisant ainsi l'espace de représentation pour l'utilisateur. Nous pensons que cette représentation augmente l'immersion de l'utilisateur dans l'environnement 3-D par rapport aux *MT-viewport*. De manière similaire à la technique précédente, l'utilisateur peut tirer profit de la manipulation directe pour contrôler la position de l'objet dans le plan de l'écran. Un mouvement de va-et-vient du second doigt permet le contrôle de la position en profondeur.

Cette technique est semblable à celle proposée par Venolia [103] où la souris est utilisée pour le positionnement en profondeur. La *Z-technique* est également analogue à la technique *Balloon Selection* [6] : le contrôle de la profondeur est effectué par un doigt séparé. La figure 3.4 représente la technique dans la taxonomie précédemment introduite. Nous allons par la suite détailler chacun des modes.

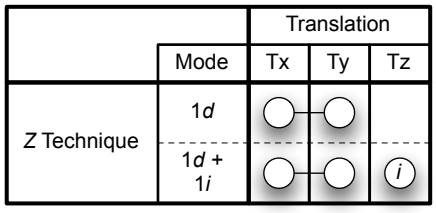


FIGURE 3.4 – Illustration de la technique *Z-technique* dans notre taxonomie.

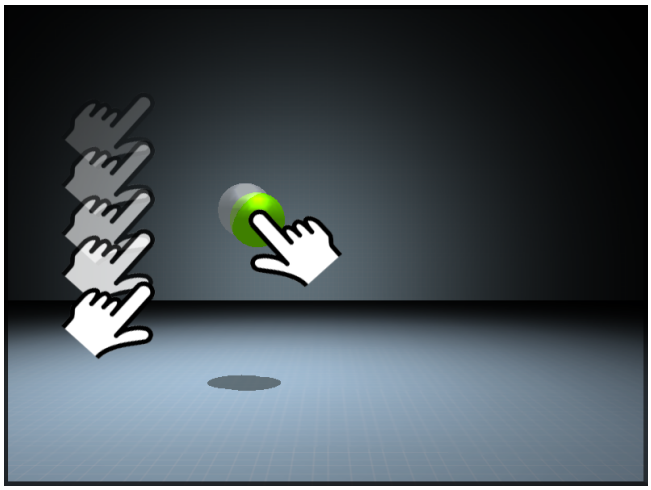


FIGURE 3.5 – Capture d'écran de la *Z-technique*. Le doigt de droite est utilisé pour un positionnement de manière directe dans le plan de la caméra tandis que le doigt de gauche est utilisé pour un positionnement indirect de la profondeur. Les mouvements en avant et en arrière rapprochent ou éloignent l'objet de l'utilisateur.

Translation à deux degrés de liberté

Quand un premier doigt rentre en contact avec la surface, un rayon est lancé dans la scène 3-D. Le premier objet traversé par ce rayon est retourné pour débuter la manipulation. Dans ce mode, la translation est limitée au plan défini par la caméra, de telle sorte que chaque mouvement de doigt résulte en un mouvement similaire de l'objet. Nous nous situons dans de l'interaction directe dans ce mode. En contrôlant les deux degrés de liberté de manière intégrale, l'utilisateur a l'impression que l'objet reste *collé* sous son doigt.

Translation à trois degrés de liberté

Lorsque l'utilisateur pose un second doigt sur la surface, nous mesurons son mouvement relatif et utilisons les mouvements de va-et-vient pour contrôler la position en profondeur. Ces mouvements de va-et-vient sont mesurés relativement à l'utilisateur. Les mouvements vers l'avant *éloignent* l'objet de l'utilisateur tandis que les mouvements vers l'arrière *rapprochent* l'objet (figure 3.5). Comme nous mesurons un mouvement relatif, il est possible d'utiliser une succession de mouvements du doigt ; principe également connu sous le nom de débrayage ; pour positionner l'objet à la profondeur voulue [31].

Durant les tests préliminaires, nous nous sommes aperçu qu'il était parfois difficile d'obtenir un positionnement précis dû principalement à une faible résolution de suivi du doigt. Comme le contrôle de la profondeur s'effectue d'une manière indirecte (figure 3.4), nous avons donc choisi d'utiliser une fonction de transfert pour relier le mouvement du doigt au déplacement de l'objet. Nous avons opté pour une fonction non linéaire qui relie la vitesse de déplacement du doigt à un facteur d'échelle. Une vitesse de déplacement réduite correspond à un facteur d'échelle faible, alors qu'un déplacement rapide correspond à un facteur beaucoup plus élevé. Cela permet d'obtenir à la fois un déplacement fin et un

déplacement grossier tout en maintenant un effort de déplacement du doigt minimal. Les valeurs de la fonction de transfert furent évaluées et testées par deux utilisateurs experts de la technique de manière à optimiser ce lien entre mouvement des doigts et mouvements de l'objet [33, 30].

3.3 Évaluation des techniques

3.3.1 Objectifs

Les deux techniques présentées sont similaires dans la manière de contrôler l'objet dans le plan de la caméra. En effet, elles utilisent toute les deux le positionnement de manière directe : l'utilisateur touche l'objet et celui-ci reste sous son doigt. Cependant, les deux techniques diffèrent dans la manière de présenter les informations de la scène 3-D ainsi que dans le contrôle de la profondeur. Les *MT-viewport* présentent l'information avec quatre vues de la scène et le positionnement de la profondeur s'effectue de manière directe. La *Z-technique* procure une visualisation plein-écran et contrôle la profondeur de manière indirecte.

Par l'évaluation de ces techniques, nous voulons donc étudier l'impact de la présentation des informations ainsi que la manière de contrôler la profondeur (i.e. directe ou indirecte). Nous nous intéresserons à la fois aux effets sur les performances en terme de temps de réalisation et de coordination, ainsi qu'aux retours qualitatifs des utilisateurs. La *Z-technique* encourage le contrôle simultané des trois degrés de liberté alors que l'approche des *MT-viewport* semble favoriser un contrôle itératif dans chacune des vues de la scène, ceci malgré notre extension multipoint. Présentant une seule et unique vue de la scène 3-D, nous nous attendons également à ce que la *Z-technique* améliore le sentiment d'immersion dans l'environnement 3-D. En effet, les *MT-viewport* affichent les informations sur quatre vues, nécessitant une plus forte concentration de l'utilisateur. Outre les différences inhérentes aux techniques, nous voulons également évaluer l'influence de la précision requise pour le positionnement comme variable de contrôle.

3.3.2 Matériel

Cette expérience contrôlée a été réalisée sur une table tactile Immersion iLight [52] basée sur le principe d'illumination diffuse (i.e. *diffused illumation* dans sa dénomination originale). La table est constituée d'un écran de projection mono-scopique d'une taille de 100×70 cm (diagonale de 42 pouces) situé à une distance de 105 cm du sol. Le vidéo projecteur situé sous la table fonctionne à une fréquence de 60 Hz avec une résolution de 1400×1050 points, et procure ainsi une densité par point de 14 points par centimètres (36 DPI). Une caméra, située sous la table, fonctionnant à une fréquence de 120 Hz avec une résolution de 640×480 points, enregistre les mouvements des doigts sur la surface. Cette configuration du dispositif offre donc une résolution de 6.4 points par centimètres (16.25 DPI) pour le suivi des doigts.

Le logiciel utilisé pour le suivi et la détection des doigts était celui fourni par Immersion, iLight framework dans sa version 1.6. Les données concernant les doigts étaient

ensuite envoyées en utilisant le protocole TUIO [97] à une application 3-D réalisée par nos soins utilisant le framework Ogre3D [81].

3.3.3 Tâche réalisée

La tâche à réaliser était un positionnement 3-D basé sur une tâche de positionnement initialement exposée par Zhai [117] (figure 3.5). La tâche demandée aux utilisateurs était de positionner aussi vite que possible une sphère bleue dans une sphère grise transparente. Lorsque la sphère bleue était bien positionnée, sa couleur changeait et devenait verte. Ce retour visuel était également utilisé pour aider les participants à manipuler des objets 3-D [117].

Chaque essai de l'expérience commençait lorsque la cible précédente était bien positionnée et terminait avec le bon positionnement de la cible courante. Un essai d'expérience était considéré comme terminé lorsque la distance entre les deux centres des sphères restait sous un seuil prédéfini pendant une durée de 0.8 seconde. La sphère transparente disparaissait ensuite pour réapparaitre à une autre position tandis que la sphère bleue était repositionnée au centre de l'écran. En dehors de la perspective et de l'occlusion, nous avons ajouté un sol où était projeté les ombres des objets, de manière à améliorer la perception de la profondeur pour les utilisateurs. La caméra virtuelle restait fixe durant la durée de l'expérience.

3.3.4 Participants

Huit individus de sexe masculin avec un âge moyen de 24 ans participèrent. Sept participants étaient droitiers, un était gaucher et tous possédaient une vision normale ou corrigée. Les sujets avaient une expérience variable avec les applications de réalité virtuelle et les applications 3-D mais, comme nous étudions un comportement moteur de bas niveau, cela reste acceptable. La moitié des participants étaient des utilisateurs réguliers d'une table tactile. Les autres étaient familiers avec la technologie tactile, comme les téléphones ou PC tactiles, mais n'avaient jamais utilisé de surfaces multipoints.

3.3.5 Plan expérimental

Un plan expérimental intra-sujets et une analyse à mesures répétées ont été utilisés. Trois variables indépendantes étaient présentes : Technique (*Z-technique* ou *MT-viewport*), Position de la sphère de destination avec 17 positions réparties de manière uniforme sur une demi sphère, et Précision avec 2 niveaux (faible et haute précision).

Les participants durent mener à bien 5 Bloc successifs d'essais d'expérience. Chaque Bloc était composé de 68 essais : 2 répétitions de 34 combinaisons Position × Précision. L'ordre de présentation de la variable Technique pour les participants était contrebalancé. Une pause était encouragée tous les 17 essais.

Avant de débuter l'expérience, les participants disposaient de 10 minutes d'entraînement pour s'exercer sur les différentes techniques. L'évaluation se terminait par un questionnaire qualitatif sur les ressentis de l'utilisateur. En moyenne, l'expérience durait 45 minutes.

Pour résumer, le plan expérimental était le suivant :
8 participants ×
2 TECHNIQUE ×
5 BLOC ×
2 PRÉCISION ×
17 POSITION ×
2 répétitions
= 5440 essais d'expérience.

3.4 Résultats et discussion

Nous orienterons la présentation des résultats autour des variables dépendantes de notre expérience. Ces variables seront subdivisées en trois catégories. Premièrement nous nous intéresserons au temps de positionnement, indicateur de performance. Nous examinerons ensuite plusieurs variables caractérisant la coordination ainsi l'efficacité du mouvement. Ces indicateurs nous permettront d'analyser la perception de la tâche par l'utilisateur, de même que la stratégie utilisée pour atteindre le but. Nous utiliserons pour cela la *NDC* [102], la $\boldsymbol{m}$-metric [74], ainsi que de la coordination telle que définie par Zhai [117], métriques présentées dans l'état de l'art à la section 1.6. Finalement, nous étudierons une variable spécifique à l'utilisation des *MT-viewport* : la fréquence d'utilisation simultanée de deux viewports. Cette variable contribuera à évaluer l'utilisation de notre extension multipoint.

3.4.1 Temps de positionnement

Le temps de positionnement est défini pour la tâche comme étant le temps nécessaire pour positionner avec succès la sphère bleue dans la sphère grise depuis le précédent positionnement correctement réalisé.

Une analyse à mesures répétées de la variance (ANOVA) a révélé un effet significatif de la variable BLOC ($F_{3,21} = 5.68$, $p = 0.016$) mettant en avant la présence d'un effet d'apprentissage. Une comparaison deux à deux a décelé une différence significative ($p < 0.05$) entre le premier bloc et les autres. Par conséquent, nous avons enlevé le premier bloc pour les analyses suivantes.

Nous n'avons pas trouvé de différences significatives au seuil de risque de 5% pour la variable TECHNIQUE sur le temps de positionnement. Le temps moyen est de 3.10s pour la *Z-technique* et de 3.03s pour les *MT-viewport*. Afin de pouvoir affirmer qu'aucune différence supérieure ou égale à 5% n'existait, nous avons également mené une analyse de puissance. Celle-ci a révélé que notre expérience pouvait détecter une taille d'effet de l'ordre de 0.75, ce qui représente une différence entre les techniques égale à 5%. Nous pouvons donc affirmer avec confiance qu'aucune différence supérieure ou égale à 5% n'existe entre les deux techniques pour le temps de positionnement.

L'analyse ANOVA a cependant trouvé un effet significatif de la variable PRÉCISION ($F_{1,7} = 330$, $p < 0.001$) sur le temps de positionnement. Comme nous l'attendions, une haute précision augmente de manière significative le temps de positionnement, passant

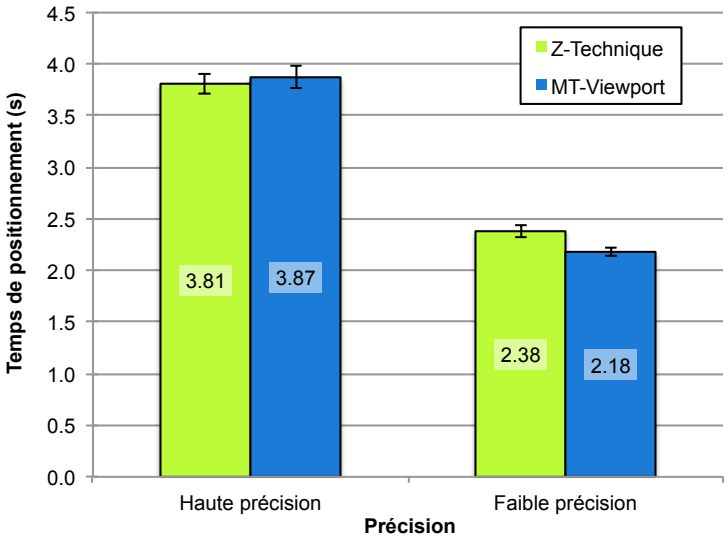


FIGURE 3.6 – Temps de positionnement moyen pour chaque technique et précision. Les barres d'erreur représentent l'intervalle de confiance à 95%.

de 2.28s pour le niveau de faible précision à 3.86s pour le niveau de haute précision. Ce résultat était cependant aisément anticipable : plus nous exigeons de précision, plus il est normal que la tâche soit difficile à réaliser. Plus intéressant, l'ANOVA a trouvé une interaction significative entre TECHNIQUE × PRÉCISION ($F_{1,7} = 6.38$, $p = 0.039$) (voir figure 3.6). Pour le niveau de faible précision, la *Z-technique* est significativement plus lente (2.38s) que les *MT-viewport* (2.18s) ($p = 0.026$). Les *MT-viewport* sont donc plus efficaces que la *Z-technique* pour ce niveau de précision.

Enfin, l'ANOVA a trouvé un effet significatif de la variable POSITION ($F_{16,112} = 8.11$, $p < 0.001$) sur le temps de positionnement. Ce résultat est cependant sans surprise : il nous révèle que le temps de positionnement est plus rapide pour les cibles ne nécessitant aucun changement de profondeur (i.e. seulement deux degrés de liberté à manipuler).

En conclusion, nous avons montré que les deux techniques sont égales au seuil de risque 5% avec néanmoins un effet de la variable PRÉCISION qui donne l'avantage aux *MT-viewport* pour le niveau de faible précision. Afin de comprendre ce résultat, analysons maintenant la coordination afin de comprendre la stratégie utilisée par les utilisateurs pour réaliser la tâche demandée.

3.4.2 Coordination

Dans cette sous-section, nous étudierons la manière dont les utilisateurs ont effectué l'évaluation, en nous attardant particulièrement sur la stratégie de manipulation des différents degrés de liberté. Tout d'abord, nous analyserons les modes de la taxonomie utilisés, afin d'obtenir une vision globale. Ensuite, nous nous intéresserons au nombre de degrés de liberté manipulés simultanément. Pour évaluer cette information, nous utiliserons la *NDC* [102]. Suivant ce résultat de *NDC*, nous adopterons différentes stratégies. En cas de manipulation d'un unique degré de liberté (i.e. $NDC = 1$) nous identifierons précisément

quel était ce degré de liberté. En cas de manipulation simultanée de multiples degrés de liberté, nous userons de la $\boldsymbol{m}$-metric pour déterminer les couples manipulés. Enfin, nous évaluerons l'efficacité quantitativement avec la coordination telle que définie par Zhai [117] et qualitativement avec la trajectoire.

Temps passé dans les différents modes

Pour obtenir une vision globale des degrés de liberté manipulés, nous avons commencé par calculer le temps moyen passé dans chacun des modes présentés dans les taxonomies. Les résultats, exposés dans le tableau 3.1, dévoilent un comportement totalement différent suivant les techniques.

Intitulé	*Sans Précision*	*Avec Précision*	*Moyenne*
Mode 0	1%	1%	1%
Mode $1d$	84%	92%	88%
Mode $1d + 1d$	15%	7%	11%

(a) Technique *MT-viewport*

Intitulé	*Sans Précision*	*Avec Précision*	*Moyenne*
Mode 0	1%	1%	1%
Mode $1d$	24%	25%	25.5%
Mode $1d + 1d$	75%	74%	74.5%

(b) *Z-technique*

TABLE 3.1 – Pourcentage de temps passé dans les différents modes.

La technique *MT-viewport* semble favoriser la séparation des degrés de liberté. En effet, les utilisateurs passent 88% du temps dans le mode $1d$ (contrôle de 2 degrés de liberté), en comparaison avec le mode $1d+1d$ (contrôle des 3 degrés de liberté) actif seulement 11% du temps. Une analyse ANOVA a trouvé un effet significatif de la variable PRÉCISION sur le temps passé dans les modes $1d$ et $1d+1d$ ($p = 0.02$). Pour cette technique, nous remarquons que l'augmentation du niveau de précision entraine un changement de comportement de la part de l'utilisateur en ce qui concerne les modes.

Pour la *Z-technique*, les résultats indiquent un comportement différent. L'analyse ANOVA n'a pas trouvé d'effet significatif pour la variable PRÉCISION. Peu importe le niveau de difficulté de la tâche, l'utilisateur semble manipuler la technique de la même manière. Dans sa globalité, la technique semble favoriser le contrôle intégral des degrés de liberté, les utilisateurs passant en moyenne 74.5% du temps dans le mode $1d + 1i$ (contrôle des 3 degrés de liberté). Ces tendances doivent néanmoins être approfondies. En effet, activer le mode $1d + 1i$ ne signifie pas forcément contrôler 3 degrés de liberté. L'utilisateur peut, par exemple, activer le mode mais contrôler uniquement Tz, laissant Tx et Ty constants.

Afin de lever ces ambiguïtés, nous allons maintenant analyser le nombre de degrés de liberté manipulés par l'utilisateur à l'aide de la *NDC*.

Nombre de degrés de liberté manipulés

Dans un premier temps, nous allons nous intéresser au pourcentage de temps passé à manipuler un ou plusieurs degrés de liberté. Les résultats de la *NDC* sont présentées dans la tableau 3.2.

Technique	NDC1	NDC2	NDC3
Z-technique	50%	44%	6%
MT-viewport	66%	33%	1%

TABLE 3.2 – Résultat de la *NDC* pour les différentes techniques en pourcentage du temps global.

Pour les 3 valeurs de la *NDC*, l'analyse ANOVA a trouvé des différences significatives entre les techniques (NDC1 $F_{1,7} = 140.23$, NDC2 $F_{1,7} = 113.61$, NDC3 $F_{1,7} = 145.15$, $p < 0.001$ pour les trois NDC). Ces différences renforcent notre hypothèse tirée de l'analyse des modes : la *Z-technique* favorise davantage la manipulation de plusieurs degrés de liberté que la technique *MT-viewport*.

L'ANOVA spécifique à NDC3 révéla également un effet significatif de la variable BLOC ($F_{3,21} = 5.39$, $p < 0.01$). Une comparaison deux à deux nous indiqua que le bloc 1 présentait une différence significative avec les autres ($p < 0.01$). Cela illustre un apprentissage de l'utilisateur pour manipuler simultanément plusieurs degrés de liberté.

L'étude statistique de la variable PRÉCISION nous dévoila cependant des nuances quant à cette manipulation simultanée des degrés de liberté. En effet, une interaction entre les variables TECHNIQUE × PRÉCISION était présente pour NDC1 et NDC2 (MDS1 $F_{1,7} = 17.90$, NDC2 $F_{1,7} = 21.88$, $p < 0.005$ pour les deux NDC). L'analyse détaillée illustre un comportement différent pour les deux techniques.

Pour la technique *MT-viewport*, l'utilisateur augmente significativement ($p < 0.001$) la manipulation d'un seul degré de liberté passant de 61% à 71% lorsqu'une grande précision est nécessaire. En conséquence, nous notons une baisse significative ($p < 0.001$) de la manipulation de deux degrés de liberté passant de 38% à 29%.

Pour la *Z-technique*, l'augmentation de NDC1 reste présente mais la différence entre les 2 niveaux devient non plus significative mais marginale ($p = 0.05$). La manipulation d'un degré de liberté passe de 48% dans le niveau de faible précision à 51% dans l'autre niveau. Pour la NDC2, le changement de précision ne semble pas influer le comportement de l'utilisateur, la comparaison deux à deux n'ayant fournie aucune différence significative.

Ces analyses corroborent nos hypothèses de départ sur l'utilisation des techniques. La technique *MT-viewport* incite l'utilisateur à effectuer ses mouvements de manière séquentielle, manipulant fréquemment un seul degré de liberté à la fois. Cette conclusion est particulièrement valable lorsque la tâche demandée nécessite un haut niveau de précision. Pour la *Z-technique*, l'utilisateur semble tirer avantage de la visualisation plein écran en manipulant plus fréquemment de multiples degrés de liberté. La variable de contrôle PRÉCISION révéla également un effet lié à la visualisation plein écran. Lorsque la tâche nécessite une haute précision, l'utilisateur ne montre pas de modification importante de comportement avec la *Z-technique*, en comparaison avec la technique *MT-viewport*. Nous pensons que cela dérive de l'espace physique de manipulation, en plein écran pour la

Z-technique. Ce sont néanmoins des résultats préliminaires et nous allons maintenant identifier et analyser précisément les degrés de liberté manipulés afin de consolider ces hypothèses.

Identification des degrés de liberté manipulés séparément

Nous allons commencer par déterminer quels sont les degrés de liberté manipulés de manière séparée. Pour cela, nous avons utilisé le résultat de la NDC1 à partir duquel nous avons extrait précisément les degrés de liberté utilisés. Les résultats sont présentées dans le tableau 3.3. Les deux techniques ont des comportements différents, qui reflètent leurs taxonomies.

Technique	Tx	Ty	Tz	Total (NDC1)
Z-technique	13%	8%	30%	50%
MT-viewport	30%	11%	25%	66%

TABLE 3.3 – Pourcentage de temps passé à manipuler un degré de liberté unique.

Pour la technique *MT-viewport*, nous remarquons une prédominance de deux degrés de liberté, Tx et Tz. Ce résultat nous procure des précisions sur le comportement des utilisateurs : ceux-ci semblent manipuler plus fréquemment les viewports situé en bas de l'écran (figure 3.4). Une analyse plus précise de l'utilisation des viewports nous indiqua qu'effectivement, les viewports proches de l'utilisateur étaient utilisés pendant 99.5% du temps. Ce résultat est en adéquation avec la taxonomie et nous laisse envisager une utilisation différente des viewports. Nous pourrions par exemple profiter de cet espace pour afficher d'autres informations ou adapter la technique à des écrans de taille plus réduite.

Pour la *Z-technique*, le degré de liberté Tz, contrôlé de manière séparée dans la taxonomie, prédomine. Cela indique une tendance de l'utilisateur à le contrôler séparément. Ce résultat semble néanmoins logique : en proposant un contrôle séparé dans la taxonomie, il était normal de s'attendre à ce que l'utilisateur sépare ce degré de liberté. Toutefois, cela nous procure des informations intéressantes sur l'utilisation du mode $1d + 1i$, activé 74% du temps. Malgré la possibilité de paralléliser ses actions dans ce mode, l'utilisateur semble utiliser une stratégie séquentielle pendant une large période de temps.

L'ANOVA révéla une interaction entre TECHNIQUE × PRÉCISION, significative pour Tx et Ty (Tx $F_{1,7} = 17.80$, Ty $F_{1,7} = 31.72$, $p < 0.005$ pour les deux) et marginale pour Tz ($F_{1,7} = 5.81$, $p < 0.05$). Une comparaison deux à deux nous dévoila un effet spécifique à la technique *MT-viewport*. Lorsqu'une grande précision est demandée, l'utilisateur augmente significativement la manipulation de Tx, Ty et Tz ($p < 0.003$). A nouveau, nous notons que plus la tâche devient difficile, plus l'utilisateur à tendance a séparer la manipulation des degrés de liberté pour la technique *MT-viewport*.

Pour la *Z-technique*, les résultats de l'ANOVA sont légèrement différents. Tout d'abord, la comparaison deux à deux n'a trouvé aucun effet pour la variable PRÉCISION sur la manipulation de Tx et Ty. Concernant Tz, l'analyse statistique a trouvé une interaction entre TECHNIQUE × BLOC ($F_{1,7} = 3.38$, $p = 0.035$). Le premier bloc étant significativement ($p = 0.02$) plus élevé que les autres, nous observons donc un effet d'apprentissage

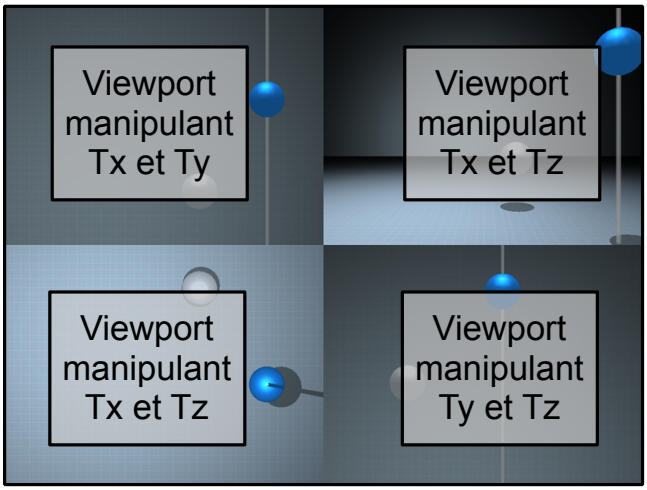


TABLE 3.4 – Les différents viewports et les degrés de liberté associés pour la technique *MT-viewport*.

de la part de l'utilisateur sur la manipulation de Tz. Ce résultat entre en adéquation avec l'effet d'apprentissage trouvé pour NDC3. Malgré une tendance initiale à séparer le contrôle des degrés de liberté, l'utilisateur apprend à contrôler les degrés de liberté de manière intégrale avec la *Z-technique*. Cependant, son comportement change en fonction de la précision demandée par la tâche. Tout comme pour la technique *MT-viewport*, la variable PRÉCISION influe significativement ($p < 0.001$) la manipulation de Tz pour la *Z-technique*. Celle-ci passe de 27% dans le niveau de faible précision à 31% dans l'autre niveau. Lorsqu'une grande précision est nécessaire, l'utilisateur adapte sa stratégie de manipulation résultant en une séparation plus importante de Tz par rapport aux deux autres degrés de liberté.

Identification des degrés de liberté manipulés intégralement

Nous avons précédemment étudié la manipulation des degrés de liberté manipulé séparément. Nous allons maintenant nous intéresser aux degrés de liberté manipulés intégralement. Pour cela nous allons utiliser la $\boldsymbol{m}$-metric [74], introduite dans l'état de l'art (section 1.6.4). Nous allons ensuite relier ces résultats avec les modes de la taxonomie afin de déterminer la manière dont sont utilisés les modes. La tâche de positionnement nécessitant le contrôle de trois degrés de liberté, nous allons donc étudier la $\boldsymbol{m}$-metric pour quatre couples : (Tx, Ty), (Tx, Tz), (Ty, Tz) et enfin (Tx, Ty, Tz). Les moyennes sont illustrées dans la figure 3.7.

Commençons par la technique *MT-viewport*. Immédiatement, nous observons que deux couples se dégagent du lot : (Tx, Tz) et (Ty, Tz). Ces couples correspondent à l'utilisation des viewports 2 et 3 situés en bas de l'écran (cf tableau 3.4). Ce résultat précise ce que nous avions précédemment révélé : les viewports proches de l'utilisateur sont majoritairement utilisés, particulièrement le viewport de gauche. Le très faible score du couple (Tx, Ty, Tz) nous informe également que l'utilisateur n'a pas profité de la possibilité offerte par la technique à contrôler simultanément les trois degrés de liberté. Les 13% de temps passé

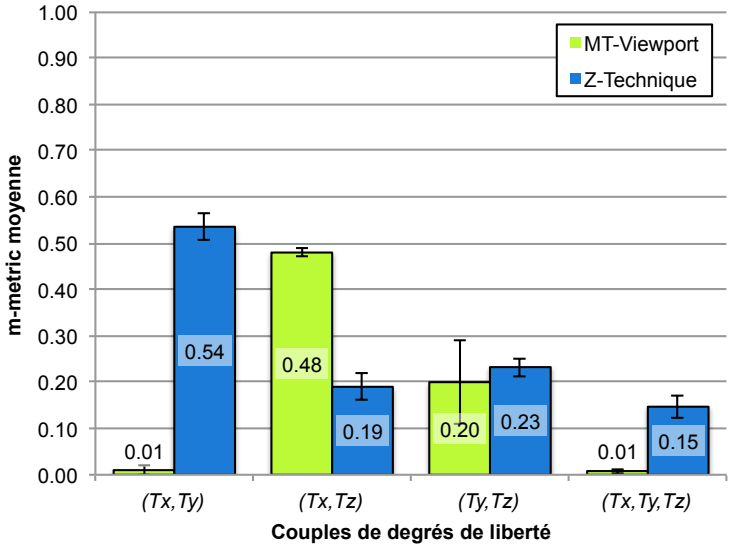


FIGURE 3.7 – $\boldsymbol{m}$-metric moyenne pour chaque technique. Les barres d'erreur représentent l'intervalle de confiance à 95%.

dans le mode $1d + 1d$ ont donc été consacré à un contrôle séparé des degrés de liberté, précédemment illustré par le fort pourcentage de temps passé à manipuler Tx et Tz.

Pour la *Z-technique* le profil de la $\boldsymbol{m}$-metric nous apporte des informations intéressantes sur la manière dont l'utilisateur a coordonné ses mouvements. Tout d'abord, le couple (Tx, Ty) se distingue très fortement, illustrant les 26% du temps passé dans le mode $1d$. Les autres couples nous indiquent que l'utilisateur manipulait le degré de liberté Tz en combinaison avec Tx ou Ty.

Ces résultats fournissent des informations sur l'influence de la présentation plein écran. En comparaison avec la technique *MT-viewport*, les utilisateurs ont été plus enclin à utiliser leurs deux mains simultanément avec la *Z-technique*. Nous pensons que ce résultat provient de la manière dont est présentée l'information, l'utilisateur devant uniquement porter son attention sur un seul endroit de l'écran avec la *Z-technique*. Cette hypothèse est également corroborée avec les scores du couple (Tx, Ty, Tz), significativement plus important avec la *Z-technique*.

L'ANOVA montra également une interaction entre TECHNIQUE × PRÉCISION significative pour le couples (Tx, Ty) ($F_{1,7} = 9.76$, $p = 0.015$) et marginale pour le couple (Tx, Ty, Tz) ($F_{1,7} = 5.27$, $p = 0.05$). La comparaison deux à deux révéla un comportement propre à la *Z-technique*. L'augmentation de la précision résulte en une diminution significative des couples (Tx, Ty) ($p = 0.002$) et (Tx, Ty, Tz) ($p < 0.03$). Nous retrouvons une nouvelle fois un comportement différent lors d'un haut niveau de précision.

Efficacité et trajectoire 3-D

Les précédentes analyses nous ont permis de dévoiler les degrés de liberté manipulés seul ou simultanément, sans apporter de réponse à la question d'efficacité sur le mouvement global. Nous allons maintenant étudier l'efficacité telle que décrite par Zhai [116]

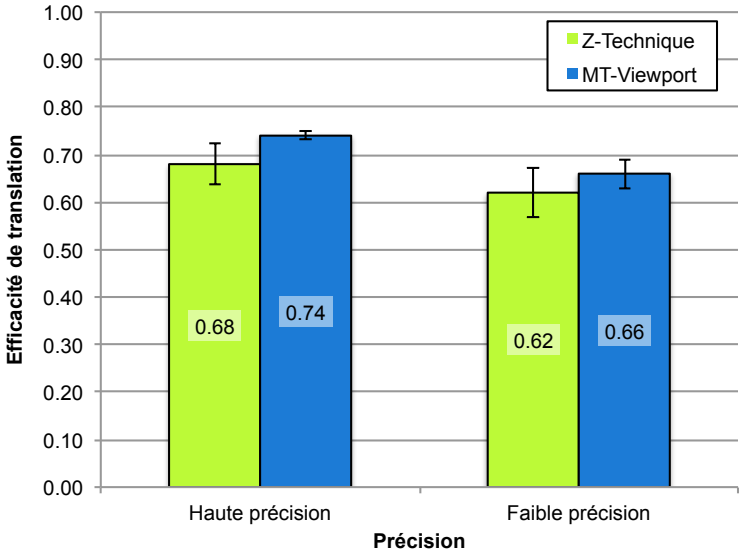


FIGURE 3.8 – Efficacité moyenne pour chaque technique et précision. Les barres d'erreur représentent l'intervalle de confiance à 95%.

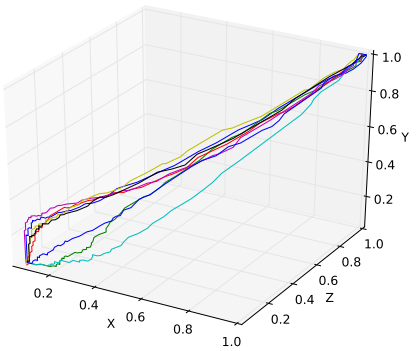
ainsi que procéder à une évaluation qualitative des courbes de trajectoires pour les deux techniques.

L'efficacité est définie comme le ratio de la longueur du chemin le plus court par la longueur du chemin effectivement parcouru du début (i.e. sphère au centre de l'écran) à la fin (i.e. sphère correctement positionné) d'un essai d'expérience [116].

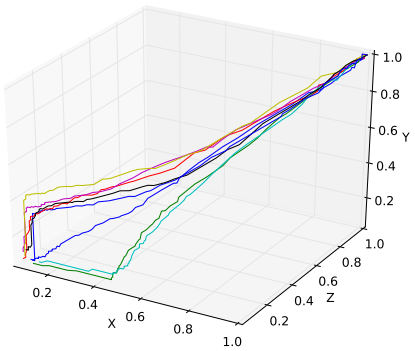
L'analyse ANOVA a trouvé un seul et unique effet, celui de la variable PRÉCISION ($F_{1,7} = 42.47$, $p < 0.001$) illustrant le fait que les utilisateurs étaient beaucoup plus efficaces pour les cibles nécessitant une faible précision (figure 3.8). L'efficacité moyenne est de 0.64 pour la *Z-technique* et de 0.71 pour *MT-viewport*. Ce résultat est en adéquation avec les valeurs de coordination ($\boldsymbol{m}$-metric) trouvées dans les précédentes sous-sections.

Tandis que ce résultat nous donne une information quantitative sur la globalité du mouvement produit par l'utilisateur, nous allons maintenant analyser la trajectoire effectuée par la sphère afin d'obtenir une information qualitative. Nous voulons ainsi observer le comportement de l'utilisateur sur la durée de la tâche, afin de distinguer d'éventuelles phases de manipulation comme illustrées par Wang [107] pour des objets réels. Pour chaque utilisateur, nous allons tracer une courbe 3-D qui représentera la position de la sphère au cours du temps. Afin de pouvoir comparer les différents utilisateurs, nous normaliserons ces courbes. Elles débuteront aux coordonnées $(1, 1, 1)$ et finiront aux coordonnées $(0, 0, 0)$.

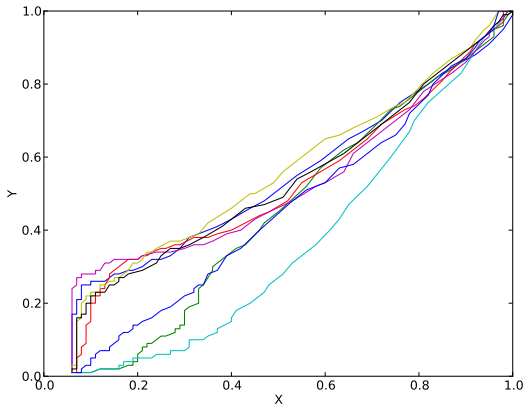
Commençons par les courbes de la technique *MT-viewport* (figure 3.9). Nous avons tracé deux courbes afin d'analyser l'influence de la variable PRÉCISION sur le mouvement. Dans le niveau de faible précision (figure 3.9b), nous observons deux stratégies principales, illustrant les remarques précédentes sur l'utilisation des viewports proches de l'utilisateur. Chaque stratégie est similaire et se déroule en deux temps. Premièrement, une phase de manipulation (mode $1d$) dans un plan correspondant aux viewports du bas (plans xOz ou yOz). Puis, une fois que la sphère a atteint la bonne position dans ce plan, une phase



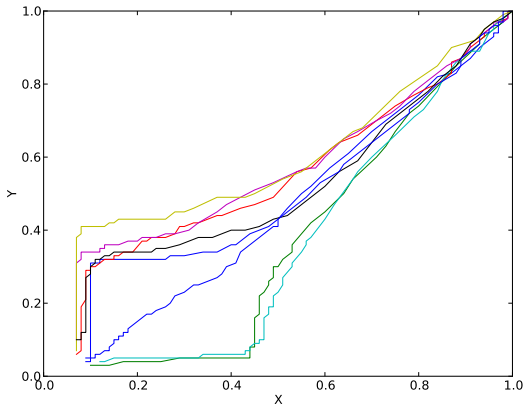
(a) 3-D avec besoin de précision.



(b) 3-D sans besoin de précision.



(c) Sur le plan XY avec besoin de précision.

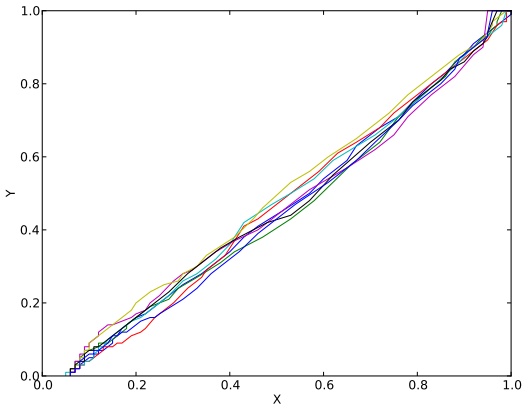


(d) Sur le plan XY sans besoin de précision.

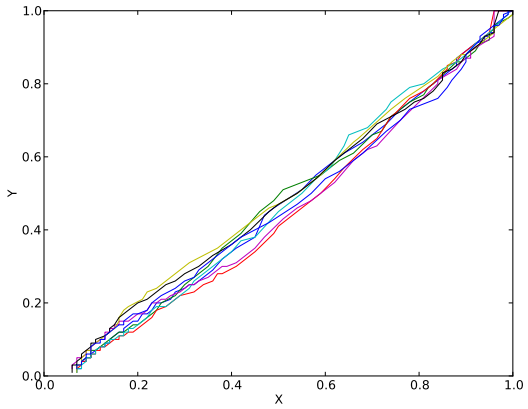
FIGURE 3.9 – Trajectoires 2-D et 3-D des différents participants pour la technique *MT-viewport*.

de manipulation (mode $1d + 1d$) pour ajuster le dernier degré de liberté séparément, en utilisant les possibilités de téléportation offertes dans ce mode (figure 3.9d). Pour plus de détails, les courbes de projections sur les différents axes se trouvent dans l'annexe A. Lorsque une haute précision est demandée à l'utilisateur, nous ne distinguons plus ces stratégies (figure 3.9a). La phase de fin est maintenant constituée d'une multitude d'ajustements (figure 3.9d), illustrés par des courbes aux allures de marches d'escalier indiquant une manipulation successive de degrés de liberté séparés.

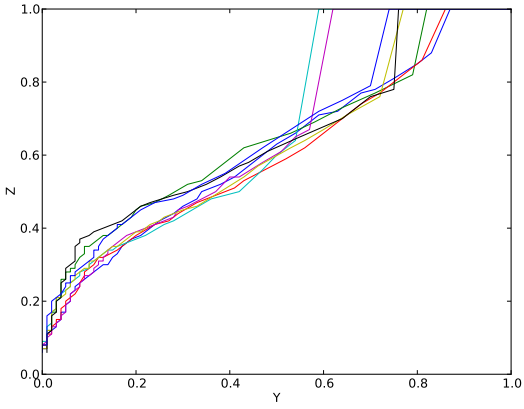
Pour la *Z-technique*, l'influence de la variable PRÉCISION est moins visible. Les deux courbes illustrent une manipulation plus intégrale des degrés de liberté, en comparaison avec la technique *MT-viewport*. L'analyse des courbes de projection illustre le bon score de la $\boldsymbol{m}$-metric pour le couple (Tx, Ty), représenté par une courbe 2-D très proche de la courbe idéal d'équation $y = x$ (figure 3.10a, 3.10b). L'observation des projections sur les plan xOz et yOz illustre un comportement en trois phases. Tout d'abord une mani-



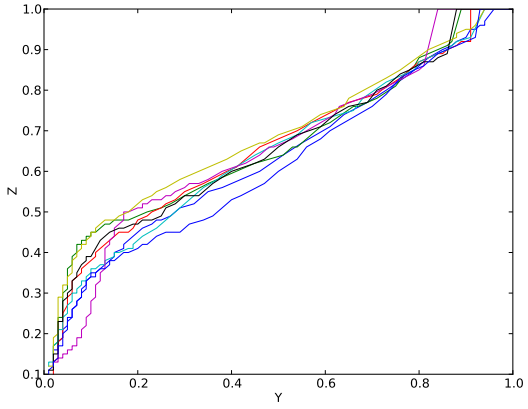
(a) Sur le plan XY avec besoin de précision.



(b) Sur le plan XY sans besoin de précision.



(c) Sur le plan YZ avec besoin de précision.



(d) Sur le plan YZ sans besoin de précision.

FIGURE 3.10 – Trajectoires 2-D des différents participants pour la technique *Z-technique*.

pulation séparée de Tx ou Ty, laissant Tz de côté. Ensuite, une phase de manipulation intégrale, alliant Tz avec Tx ou Ty. Enfin, on remarque que la manipulation de Tz devient majoritaire. L'effet de la variable PRÉCISION est également visible. Une haute précision semble retarder le début de la manipulation de Tz (figure 3.10c, 3.10d). Toutes les courbes de projection sont également présentes dans l'annexe A.

3.4.3 Utilisation de l'extension multipoint de la technique *MT-viewport*

Pour la technique *MT-viewport* nous avons analysé la fréquence d'utilisation simultanée de deux viewports. En effet, nous avons étendu la technique traditionnelle *4-viewport* mais il convient de savoir dans quelle mesure les utilisateurs tiraient profit de cette extension. L'extension fut considérée utilisée si deux viewports étaient actifs simultanément au moins une fois durant un essai d'expérience. Ces jets étaient ensuite marqués pour le

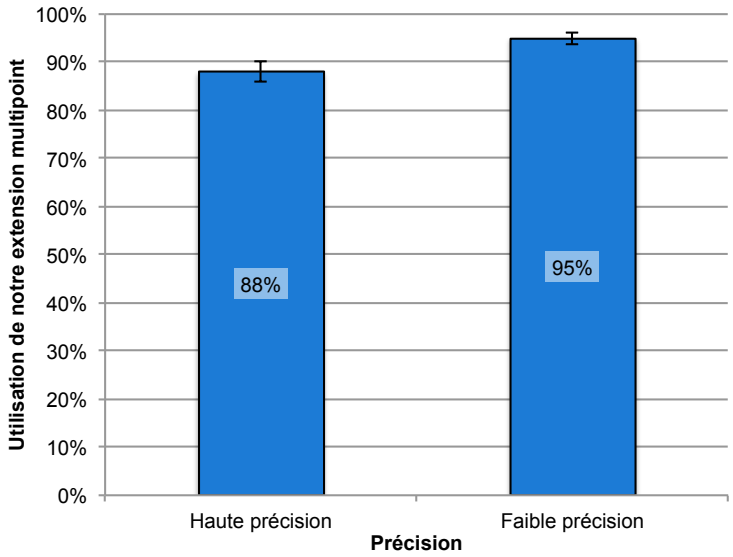


FIGURE 3.11 – Fréquence d'utilisation d'un second viewport. Les barres d'erreur représentent l'intervalle de confiance à 95%.

traitement statistique des données. En moyenne, un second viewport a été utilisé 92 fois sur 100 (figure 3.11). Ce résultat, mis en relation avec les 11% de temps passé dans le mode $1d + 1d$, nous indique que l'utilisation de notre extension était fréquente mais très limité dans la durée. Comme illustré par les courbes de trajectoire, cela indique une forte utilisation de la fonction téléportation de ce mode, peu de temps étant nécessaire pour déplacer l'objet à la position voulue.

L'analyse ANOVA a également mis en évidence un effet marginal de la variable PRÉCISION ($F_{1,7} = 5.35$, $p < 0.054$) sur l'utilisation de notre extension. Celle ci a été plus utilisée dans le niveau faible précision (95.5%) que dans le niveau haute précision (88.1%). Comme illustrée par l'analyse de trajectoire, cela nous précise que l'utilisateur tend a séparer le contrôle des degrés de liberté lorsque la tâche demande une attention plus grande.

3.4.4 Analyse qualitative

Parmi tout les participants, six ont préféré la *Z-technique*, un la technique *MT-viewport* et un n'a déclaré avoir aucune préférence. Le participant préférant la technique *MT-viewport* a trouvé la technique plus facile d'utilisation. Curieusement, ce participant avait mentionné une utilisation fréquente de logiciels de CAO, utilisant la technique traditionnelle *4-viewport*. Les participants en faveur de la *Z-technique* ont trouvé la technique plus facile à utiliser et ont particulièrement apprécié le fait d'avoir une seule vue pleine écran de la scène 3-D. L'utilisation d'un second doigt pour manipuler de manière indirecte la profondeur était, selon eux, difficile à appréhender au début mais très efficace une fois maîtrisée. Nous retrouvons ici l'effet d'apprentissage identifié durant l'analyse statistique.

Nous avons également demandé aux participants de noter leur niveau de fatigue pour chaque technique. De manière générale, aucun résultat spécifique aux techniques n'est ressorti de cette évaluation de la fatigue. Peu importe l'ordre de présentation des techniques,

les participants ont signalé leur fatigue lors de la deuxième technique, ceci malgré les nombreuses pauses proposées. La principale conclusion concerne donc l'utilisation longue durée d'une table tactile : il est fatiguant de travailler debout devant de tels dispositifs pendant une période prolongée.

Pour la *Z-technique*, tous les participants (incluant le seul gaucher de l'évaluation) ont utilisé leur main dominante pour manipuler l'objet dans le plan de la caméra, favorisant l'utilisation de leur main non dominante pour le contrôle de la profondeur. Une exception cependant, un utilisateur a choisi d'utiliser son pouce de la main dominante pour contrôler la profondeur. Deux participants ont également ennuyés par la direction choisie (i.e. mouvement de bas en haut et vice-versa) pour le contrôle de la profondeur. En tirant l'objet vers eux, ils s'attendaient à le voir s'éloigner plutôt que se rapprocher. Dans une application réelle, ce paramètre devra être pris en compte et personnalisé selon les préférences de l'utilisateur.

Une autre remarque fut reportée pour la technique *MT-viewport*. Les utilisateurs se sont plaint de la charge cognitive induite par les différents viewports. Ils jugèrent difficile la coordination de la tâche sur les différents viewports, imposant un nombre conséquent de va et vient du regard sur les différentes vues. Cela précise nos précédentes conclusions sur la séparation des degrés de liberté pour cette technique. L'utilisateur, devant se concentrer sur deux viewports, a tendance à séparer leur contrôle. Cela explique également l'influence de la variable PRÉCISION pour cette technique. Plus le besoin de précision augmente, plus l'utilisateur dois effectuer de va et vient sur l'observation des viewports afin de positionner la sphère, expliquant l'augmentation de la séparation des degrés de liberté dans ce niveau.

3.5 Conclusion

Ces résultats nous montrent que les deux techniques évaluées sont équivalentes en terme de performance car aucune différence significative n'a été trouvée pour le temps de positionnement. De petites différences, inférieures à 5% peuvent néanmoins exister. Cela reste cependant négligeable pour notre étude, l'objet de nos travaux se focalisant sur l'intégration/séparation des degrés de liberté ainsi que les stratégies utilisateurs.

Durant l'expérience, les participants ont utilisé deux viewports simultanément plus de 8 fois sur 10, utilisant de manière effective notre extension multipoint et ses fonctions. L'étude statistique nous a révélé que l'usage de l'extension était fortement lié au niveau de précision demandée, résultat confirmé par l'analyse de la stratégie utilisateur dévoilant un comportement différent suivant les différents niveaux de précision.

Nous avons également montré que les deux viewports les plus éloignés de l'utilisateur étaient rarement utilisés, illustrant le fait que les utilisateurs tendent à minimiser leur fatigue lors de la réalisation de la tâche en choisissant les viewports proches d'eux. Ce résultat permet de suggérer une extension de la technique en étendant la vue perspective à la moitié supérieure de l'écran.

Oh et al. [82] avaient démontré que la technique traditionnelle *4-viewport* était plus rapide par comparaison à des techniques plein-écran. Nous fournissons ici une alternative, la *Z-technique*, qui permet d'obtenir des performances semblables à la technique des *MT-viewport* tout en étant plus appréciée par les utilisateurs. La *Z-technique* constitue donc

une alternative viable aux *MT-viewport*, surtout lorsque la surface d'affichage se trouve réduite (e.g. sur les téléphones mobiles).

Concernant la stratégie utilisateur, nous avons démontré l'influence de la précision de la tâche sur la manipulation des degrés de liberté. Peu importe la technique, plus la tâche est difficile, plus la manipulation des degrés de liberté s'effectue de manière séparée, illustrant des mouvements séquentiels de la part des utilisateurs. Les techniques ne se comportent cependant pas de la même façon : la *Z-technique* favorise un contrôle des degrés de liberté plus intégral par comparaison avec la technique *MT-viewport*. Nous pensons que cela provient de la présentation de l'information, l'utilisateur devant concentrer son attention sur plusieurs zones de l'écran. Cela s'illustre particulièrement dans le cas où la tâche nécessite une haute précision. Dans cette condition, l'utilisateur doit effectuer un nombre de va et vient du regard plus important entre les viewports, entrainant un contrôle plus séparé des degrés de liberté. Pour la *Z-technique*, l'étude approfondie nous a permis de nuancer cet effet de la précision. L'utilisateur ne change pas son comportement global mais accentue néanmoins la séparation du degré de liberté Tz. Ce comportement en adéquation avec la taxonomie, celle-ci permettant une séparation aisée du degré de liberté.

Cette différence de comportement suivant le niveau de précision met en valeur l'importance de la façon de contrôler les degrés de liberté. La *Z-technique* permet le contrôle du degré de liberté séparé avec l'aide d'un doigt indirect. La technique *MT-viewport* quant à elle, offre le contrôle du degré de liberté séparé de manière directe. Cette distinction implique des comportements différents de la part des utilisateurs. Ceux-ci se révèlent plus enclins à effectuer des actions simultanées avec le contrôle indirect, qui permet donc une intégration du contrôle des degrés de liberté. Ce comportement est en adéquation avec les théories de Guiard [41]. La main non-dominante, ici illustrée par le doigt indirect, fixe le contexte pour la main dominante, ici illustré par le doigt direct.

Pour résumer, nous avons exploré dans ce chapitre les techniques de positionnement libre pour écrans multipoints. Nous avons introduit une nouvelle technique qui permet de positionner librement un objet dans une scène 3-D, le tout dans une seule vue pleine-écran. Nous avons également fourni une extension d'une technique traditionnelle utilisée dans les logiciels de CAO.

Mais la manipulation 3-D ne se résume pas seulement au positionnement. Nous allons maintenant intégrer la gestion de la rotation afin d'étudier le comportement de techniques permettant une manipulation complète dans l'espace 3-D.

Chapitre 4

La manipulation d'objets 3-D

Sommaire

4.1 Introduction 72
4.2 Techniques de manipulation 3-D existantes 73
4.2.1 *Sticky Tools* 73
4.2.2 *Screen-Space* 74
4.3 Introduction de la technique *DS3* 75
4.3.1 Notre démarche 75
4.3.2 Description de la technique *DS3* 77
4.4 Évaluation préliminaire 77
4.4.1 Matériel 78
4.4.2 Tâche et participants 78
4.4.3 Résultats et discussion 79
4.5 Expérience contrôlée pour l'évaluation des techniques 80
4.5.1 Objectifs 80
4.5.2 Tâche réalisée 80
4.5.3 Participants 80
4.5.4 Plan expérimental 81
4.6 Résultats et analyse 81
4.6.1 Indicateurs de difficulté 82
4.6.2 Temps de réalisation de la tâche 82
4.6.3 Coordination 83
4.6.4 Analyse qualitative 95
4.7 Discussion et conclusion 96
4.7.1 Séparation des degrés de liberté et performance 97
4.7.2 Stratégies de séparation des degrés de liberté 97
4.7.3 Stratégie d'interaction bimanuelle 98
4.7.4 Contrôle direct vs. contrôle indirect 98

4.1 Introduction

La manipulation 3-D est actuellement un défi pour les créateurs d'interfaces 3-D. Afin de manipuler un objet dans un espace 3-D, il est nécessaire de contrôler à la fois le positionnement (i.e. translations suivant les axes x, y et z) et l'orientation (i.e. rotations autour des axes x, y et z). Actuellement, en utilisant la souris et les interfaces disponibles, orienter un objet 3-D est une tâche difficile et peut prendre de dix à trente secondes [48], temps très supérieur à la manipulation d'objets réels qui est effectuée entre une et deux secondes [107].

Comme mentionné précédemment, les écrans multipoints offrent une bande passante d'entrée plus importante en autorisant de multiples contacts simultanés. Contrairement à la souris où le pointeur contrôlé est distant de l'écran, les écrans multipoints permettent une interaction directe pour l'utilisateur : il peut ainsi *toucher* les objets [89]. Le geste Rotate-Scale-Translation (RST), utilisé en manipulation 2-D, est un exemple typique de ce paradigme d'interaction [45]. Mais, tandis que la manipulation 2-D sur écrans multipoints a été largement explorée, la manipulation 3-D reste quant à elle un domaine jusqu'à présent peu étudié. La principale difficulté pour passer de la manipulation 2-D à la manipulation 3-D consiste à gérer de manière adéquate les degrés de liberté additionnels. En effet, lors de l'utilisation d'écrans multipoints, le système d'interaction doit traiter en entrée des données 2-D (points de contact sur la surface) afin de les relier aux degrés de liberté de la tâche à effectuer. Les degrés de liberté de la manipulation 3-D étant situés dans un espace 3-D, il devient alors difficile de relier efficacement les entrées utilisations (2-D) avec les attributs de l'objet (3-D).

Traditionnellement, la manipulation 3-D s'effectue à l'aide d'un clavier et d'une souris, ou nécessite l'utilisation d'un périphériques d'entrée possédant de nombreux degrés de liberté (type *flock-of-bird*). Mais pour certains périphériques tels que les tablettes tactiles, ne disposant pas par défaut de clavier ni de souris, le seul moyen d'interagir est l'écran multipoint.

Jacob [55] a étudié la relation entre la performance, le périphérique d'entrée et la structure de perception de la tâche. Comme évoqué précédemment (section 1.5.2), la structure de contrôle du périphérique d'entrée ainsi que la structure de perception de la tâche peuvent être soit intégrales soit séparables. Jacob a trouvé une relation forte entres ces deux structures : la performance est optimisée lorsque celles-ci correspondent. Cependant, tandis que des études ont prouvé que les doigts de la main ont des degrés de liberté séparables [53], la manipulation 3-D reste une tâche intégrale [40]. Le pouce, l'index et le majeur peuvent être déplacés indépendamment alors que les utilisateurs perçoivent les attributs d'un objet 3-D (i.e. position et orientation) comme un tout. Cette disparité entre la structure séparable d'un écran multipoint et la nature intégrale de la tâche à effectuer nous amène à la question d'optimisation du lien entre les deux structures.

Tandis que des techniques comme *Sticky Tools* proposent une approche orientée séparation des degrés de liberté [44], d'autres techniques comme *Screen-Space* adoptent une stratégie de contrôle intégrale [88]. Néanmoins, le manque actuel d'évaluation rend la comparaison des approches difficile à effectuer.

Dans ce chapitre, nous commencerons par détailler les approches de Hancock et al. [44] et de Reisman et al. [88] en utilisant la taxonomie présentée dans le chapitre 2. Ensuite,

nous introduirons une nouvelle technique, basée sur la séparation du contrôle des degrés de liberté de position et d'orientation. Puis, nous évaluerons ces différentes techniques dans une expérience contrôlée. En dernier lieu, nous analyserons les résultats en orientant une nouvelle fois notre réflexion sur l'intégration et la séparation des degrés de liberté ainsi que la stratégie de manipulation adoptée par l'utilisateur.

4.2 Techniques de manipulation 3-D existantes

Ayant déjà présenté quelques techniques dans le chapitre consacré à l'état de l'art (section 1.4), nous nous proposons d'effectuer ici tout d'abord un mini résumé, puis une analyse détaillée de certaines techniques permettant la manipulation 3-D.

Hancock et al. [43] ont présenté des techniques d'interaction pour manipuler des objets 3-D en utilisant un, deux et trois doigts sur des écrans multipoints. Avec l'interaction à trois doigts, les utilisateurs peuvent à la fois positionner et orienter un objet 3-D. La translation en profondeur (Tz) est proposée comme option, réalisée en calculant la distance entre deux doigts. Cette technique, intitulée *Shallow-Depth*, a été évaluée et démontrée plus rapide ainsi que plus appréciée des utilisateurs. Néanmoins la tâche d'évaluation nécessitait le contrôle de seulement cinq degrés de liberté, laissant la translation de profondeur (Tz) de côté. Afin de palier à ce manque et ainsi réaliser une manipulation 3-D complète, ils introduiront plus tard une seconde technique, nommée *Sticky Tools*. Par la suite, nous analyserons cette technique et sa stratégie de contrôle, puis procéderons à son évaluation.

Reisman et al. [88] a présenté un outil offrant un contrôle intégral des degrés de liberté nécessaires à la manipulation 3-D. Utilisant un solveur de contraintes dans un usage inédit, le but de cet outil est de reproduire le fonctionnement du geste RST ; très largement utilisé et apprécié en manipulation 2-D ; cette fois-ci en manipulation 3-D. En s'inspirant de ce geste qui a rendu populaire les périphériques d'entrée multipoints, Reisman présente ainsi une nouvelle approche de la manipulation 3-D. Dans son approche, Reisman présente différentes variantes d'utilisation. Nous nommerons *Screen-Space* la variante que nous avons retenue. Ce sera la deuxième technique que nous analyserons et évaluerons.

4.2.1 *Sticky Tools*

Cette technique nécessite l'usage de trois doigts. Chaque doigt contrôle séparément des degrés de liberté qui sont parfois intégrés ensemble. Nous avons représenté la technique en utilisant notre taxonomie sur la figure 4.1. Comme nous allons le voir, cette technique est basée sur une stratégie mélangeant à la fois l'intégration et la séparation des degrés de liberté.

Mode $1d$

Ce mode sera similaire dans toutes les techniques. En effet que ce soit les techniques *Z-technique*, *Sticky Tools* ou encore *Screen-Space*, toutes adoptent la translation dans le plan caméra afin de tirer profit de la manipulation directe offerte par le périphérique

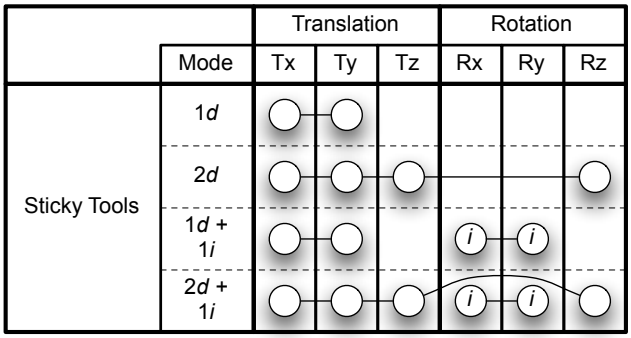


FIGURE 4.1 – Illustration de la technique *Sticky Tools* dans notre taxonomie.

d'entrée. Les deux degrés de liberté de ce mode, Tx et Ty, sont contrôlés de manière intégrale. L'utilisateur perçoit ainsi que l'objet manipulé reste *collé* sous son doigt.

Mode $2d$

Lorsqu'un deuxième doigt vient se poser sur l'objet, deux degrés de liberté supplémentaires s'ajoutent à la manipulation. Les quatre degrés de liberté de ce mode sont également contrôlés de manière intégrale. En sus de pouvoir déplacer l'objet dans le plan caméra, l'utilisateur possède maintenant la capacité d'effectuer une rotation autour de l'axe z (Rz) ainsi qu'une translation suivant cet axe (Tz). Ce mode est similaire au geste RST utilisé lors de la manipulation 2-D. En ajustant la profondeur, l'utilisateur rapproche ou éloigne l'objet de la caméra, affectant sa taille d'affichage à l'écran, similaire au changement d'échelle effectué dans le geste RST.

Autres modes

Avec l'ajout d'un doigt indirect (i.e. sans contact avec l'objet manipulé), l'utilisateur peut désormais contrôler deux degrés de liberté supplémentaires, peu importe le mode dans lequel il se trouve. Ces degrés de liberté sont contrôlés intégralement par le doigt indirect, tout en étant séparés des degrés de liberté contrôlés par le doigt direct. Un appui d'un doigt indirect dans le mode $2d$ entraine donc le passage au mode $2d + 1i$ qui permet le contrôle de tous les degrés de liberté nécessaires au positionnement et à l'orientation de l'objet.

4.2.2 *Screen-Space*

Le cœur de cette technique se situe dans l'utilisation d'un solveur de contraintes qui, fonction des coordonnées des doigts sur la surface, va ajuster la position et l'orientation de l'objet afin que les points de l'objet 3-D touchés par l'utilisateur restent *collés* sous ses doigts. Les degrés de liberté régissant la position et l'orientation sont tous contrôlés de manière intégrale par cette technique. Le rôle du solveur de contraintes consiste à minimiser l'erreur entre les points de contact dans l'espace écran et leurs correspondants dans l'espace virtuel. L'algorithme peut être synthétiser par les étapes suivantes :

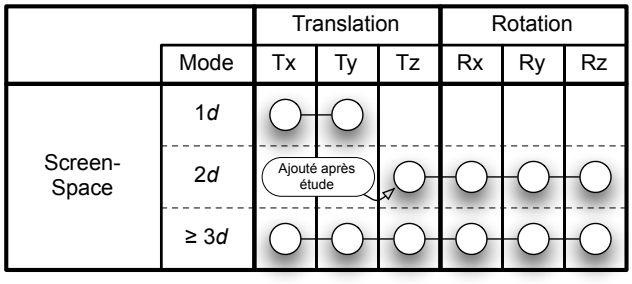


FIGURE 4.2 – Illustration de la technique *Screen-Space* dans notre taxonomie.

1. Lors d'un appui de doigt sur la surface affichant un objet 3-D :
   - Sauvegarder la position 2-D du doigt sur la surface (point $F2d_1$)
   - Sauvegarder le point 3-D associé au lancé de rayon de la position 2-D du doigt dans la scène virtuelle 3-D (point $P3d_1$)
2. Lors d'un mouvement de doigt :
   - Sauvegarder la nouvelle position 2-D (point $F2d_2$)
   - Utiliser le solveur de contraintes pour ajuster la position et l'orientation de l'objet 3-D de manière à ce que le point $F2d_2$, par lancé de rayon dans la scène 3-D, pointe vers $P3d_1$

Lorsque nous utilisons le geste RST sur une image 2-D, le même processus est utilisé mais au lieu de relier les points 2-D (i.e. les doigts) avec des points 3-D (i.e. objet dans la scène), nous relions les points 2-D (i.e. les doigts) avec des points 2-D (i.e. l'image 2-D).

Un doigt ne peut contrôler que deux degrés de liberté à l'aide de ce solveur de contraintes (nous utilisons juste la position x et y). Afin de contrôler les six degrés de liberté nécessaires à la manipulation 3-D, cette technique nécessite donc au moins trois doigts en contact direct avec l'objet. L'utilisateur peut donc contrôler au maximum deux degrés de liberté avec un doigt, quatre degrés de liberté avec deux doigts et six degrés de liberté avec trois doigts. Nous devons donc choisir les degrés de liberté à manipuler avec un et deux doigts. Nos choix sont représentés dans la figure 4.2. Avec un doigt, nous avons opté pour la manipulation directe dans le plan de la caméra, comme avec les autres techniques. Le mode $1d$ permet donc le contrôle intégral de Tx et Ty. Avec deux doigts, Reisman et al. [88] ne recommande aucun choix particulier. Nous avons donc opté pour le choix illustré par le mode $2d$, choix effectué après une expérience préliminaire que nous allons détaillée par la suite. Les degrés de liberté de ce mode sont également tous intégrés. A partir de trois doigts en contact sur l'objet, tous les degrés de liberté de position et d'orientation sont contrôlés de manière intégrale.

4.3 Introduction de la technique *DS3*

4.3.1 Notre démarche

Selon Garner [40], la structure de perception de la manipulation 3-D est composée de six degrés de liberté, perçu comme un tout par l'utilisateur. La structure de perception

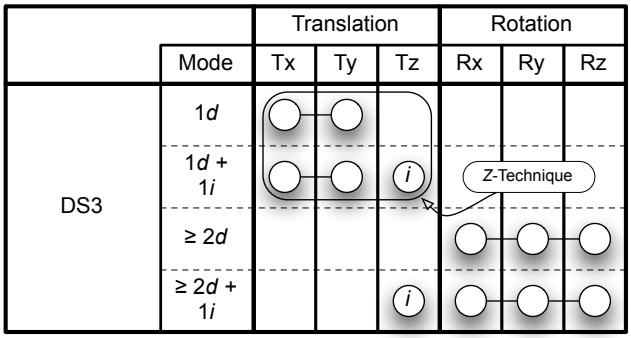


FIGURE 4.3 – Illustration de la technique *DS3* dans notre taxonomie.

est donc intégrée. Jacob [55] préconise de faire correspondre la structure de perception de la tâche avec le structure de contrôle du périphérique d'entrée. Suivre ces principes à la lettre implique donc une utilisation limitée aux périphériques d'entrée contrôlant six degrés de liberté de manière intégrale comme les souris 3-D, les baguettes 3-D ou encore les périphérique de type *flock-of-bird*.

Si l'on considère la structure séparée des doigts [53], il devient alors impossible de suivre les recommandations de Jacob et de faire correspondre la structure de perception de la tâche avec la structure de contrôle des écrans multipoints. L'état de l'art illustré précédemment ne nous apporte pas non plus de réponse claire sur la question. D'un côté nous avons *Screen-Space* qui propose le contrôle des six degrés de liberté d'une manière intégrale justifiant sa stratégie par le geste RST. De l'autre côté nous avons la technique *Sticky Tools* qui adopte une approche orientée séparation des degrés de liberté sans discuter pourquoi ni comment. Ces techniques n'ayant jamais été évaluées ni comparées, il est difficile de savoir quelle approche est favorable. Si la séparation se révèle être une approche viable, la question des degrés de liberté spécifiques à séparer se pose également.

Durant l'évaluation informelle de *Sticky Tools*, nous avons observé que le contrôle intégral de la translation et de la rotation (modes $2d$ et $2d+1i$ dans la figure 4.1) était difficile à maitriser pour les utilisateurs. Lorsque les degrés de liberté sont contrôlés séparément, notre hypothèse est qu'une séparation claire du contrôle de la position et de l'orientation améliore la manipulation pour l'utilisateur. Si nous ré-évaluons la perception de la tâche 3-D selon Garner, nous pouvons argumenter que, malgré le fait que les attributs de position et d'orientation soient perçus comme un tout, il s'agit tout de même de deux groupes d'attributs distincts que l'utilisateur peut percevoir. Dans cette définition de la tâche 3-D, séparer le contrôle de la position et de l'orientation semble alors adéquat et offre même la possibilité d'appliquer la théorie de Jacob. Par conséquent, nous avons conçu une nouvelle technique, intitulée *DS3* (*Depth-Separated Screen-Space*), basée sur cette séparation. Cette technique combine la *Z-technique* pour contrôler le positionnement, avec le solveur de contraintes introduit par Reisman [88] pour la gestion de l'orientation. La technique est représentée dans la taxonomie dans la figure 4.3

4.3.2 Description de la technique *DS3*

Mode $1d$ **et** $1d + 1i$

Ces modes reprennent la taxonomie de la *Z-technique*. Le contrôle est similaire à celui exposé dans le chapitre 3 sur le positionnement mais nous avons effectué une modification suite aux résultats et conclusions de l'expérience effectuée dans le précédent chapitre. Nous avons ainsi donné la possibilité à l'utilisateur de paramétrer le réglages du déplacement en profondeur. Ainsi, au lieu de faire des mouvements en avant pour pousser l'objet, l'utilisateur avait la possibilité d'inverser la commande afin de pouvoir pousser l'objet à l'aide de mouvement vers l'arrière.

Mode $2d$ **et** $2d + 1i$

La séparation de contrôle entre position et orientation s'effectue par le nombre de doigt en contact direct avec l'objet. Lorsque deux doigts sont présents sur l'objet, nous utilisons le solveur de contraintes pour ajuster l'orientation de l'objet, contrôlant ainsi les degrés de liberté Rx, Ry et Rz d'une manière intégrale. Nous avons également greffé le contrôle de la profondeur dans le mode $2d$. En ajoutant un doigt indirect, l'utilisateur passe au mode $2d + 1i$ qui propose l'option de contrôler le déplacement de l'objet en profondeur (Tz) de manière similaire à la *Z-technique*. Ce doigt indirect effectuant une manipulation séparée, cela ne vient pas casser la séparation de contrôle de la position et de l'orientation.

Nous venons de présenter la nouvelle technique *DS3*. Maintenant, nous allons procéder à son évaluation, en comparaison avec les techniques *Sticky Tools* et *Screen-Space*. Afin d'identifier les variables nécessaires à l'expérience contrôlée, nous avons mené une évaluation préliminaire, que nous allons détailler dans la prochaine section.

4.4 Évaluation préliminaire

Le but de cette évaluation était avant tout de tester les techniques *Sticky Tools*, *Screen-Space* et *DS3* sur une tâche concrète. Nous voulions également explorer différents choix concernant les degrés de liberté manipulés avec deux doigts pour la technique *Screen-Space*. Enfin, nous devions fixer quelques paramètres intrinsèques aux techniques.

Lors de l'utilisation de la technique *Screen-Space* avec deux doigts, nous avons la possibilité de contrôler jusqu'à quatre degrés de liberté. Afin d'éviter les translations involontaires mentionnées par Hancock [43], nous avons choisi de ne pas inclure les deux degrés de liberté qui étaient déjà contrôlés dans le mode $1d$ (Tx et Ty). Cela restreint ainsi le choix à quatre degrés de liberté. Comme exposé précédemment, nous pensons que contrôler la rotation indépendamment peut améliorer la manipulation réalisée par l'utilisateur. Nous avons donc opté pour le contrôle de trois degrés de liberté seulement, à savoir ceux responsable de l'orientation (Rx, Ry et Rz).

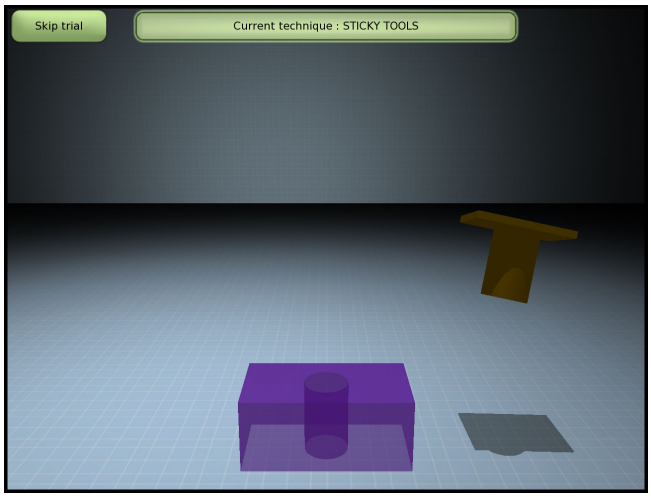


FIGURE 4.4 – Capture d'écran de la tâche *peg-in-hole*.

4.4.1 Matériel

Cette expérience contrôlée a été réalisée sur une table tactile Immersion iLight [52] basée sur le principe d'illumination diffuse (i.e. *diffused illumation* dans sa dénomination originale). La table est constituée d'un écran de projection mono-scopique d'une taille de 100×70 cm (diagonale de 42 pouces) situé à une distance de 105 cm du sol. Le vidéo projecteur situé sous la table fonctionne à une fréquence de 60 Hz avec une résolution de 1400×1050 points, et procure ainsi une densité par point de 14 points par centimètres (36 DPI). Une caméra, située sous la table, fonctionnant à une fréquence de 120 Hz avec une résolution de 640×480 points, enregistre les mouvements des doigts sur la surface. Cette configuration du dispositif offre donc une résolution de 6.4 points par centimètres (16.25 DPI) pour le suivi des doigts.

Le logiciel utilisé pour le suivi et la détection des doigts était celui fourni par Immersion, iLight framework dans sa version 1.6. Les données concernant les doigts étaient ensuite envoyées en utilisant le protocole TUIO [97] à une application 3-D réalisée par nos soins utilisant le framework Ogre3D [81].

4.4.2 Tâche et participants

La tâche à réaliser était de type *peg-in-hole* en 3-D, similaire à celle décrite par Unger et al. [98] mais sans détection de collisions (figure 4.4). Les participants devaient positionner et orienter une cheville dans un parallélépipède 3-D percé en son centre, ceci le plus rapidement possible. Le pavé 3-D était transparent afin de permettre un positionnement fin de la cheville. La cheville était constituée d'une base rectangulaire à partir de laquelle un cylindre était extrudé.

Chaque essai de l'expérience débutait lorsque la cheville *précédente* était bien positionnée et orientée et terminait quand la cheville *courante* était de nouveau bien positionnée et orientée. Lorsque l'orientation et la position de la cheville restait sous un seuil pré-déterminé, celle-ci devenait verte, indiquant à l'utilisateur qu'elle était correctement

positionnée et orientée. Un essai de l'expérience était validé lorsque la cheville restait dans une position correcte pendant une durée de 0.8 seconde. La cheville était ensuite déplacée à autre position, sélectionnée de manière aléatoire sur une demi-sphère. Le centre de la sphère était le centre du parallélépipède 3-D et le rayon était choisi de manière à occuper l'écran convenablement. Le pavé quant à lui, restait à la même position. Hormis la perspective et l'occlusion, nous avons ajouté à la scène 3-D un sol où était projetées les ombres des objets, de manière à améliorer la perception de la profondeur pour les utilisateurs. La caméra virtuelle restait fixe durant la durée l'expérience.

Dans cette évaluation préliminaire, nous avons contrôlé la présence de profondeur (i.e. si la translation suivant l'axe z était nécessaire ou non), l'axe de la rotation à effectuer (i.e. rotation nécessaire autour d'un axe unique ou d'une combinaison de plusieurs axes) et enfin l'amplitude de la rotation.

Six individus de sexe masculin (moyenne d'âge de 24 ans) ont participé. Ils possédaient une expérience variable avec la réalité virtuelle et les écrans multipoints. Deux étaient experts, un autre possédait quelques notions et les participants restants étaient novices.

4.4.3 Résultats et discussion

Nous allons présenter brièvement les résultats liés à l'évaluation préliminaire, afin de tirer les conclusions nécessaires pour la conduite de l'évaluation finale.

Commençons par le temps nécessaire pour réaliser la tâche. Celui-ci est défini comme étant le temps nécessaire pour positionner avec succès la cheville dans le pavé 3-D depuis le précédent positionnement correctement réalisé. Les premiers résultats ont mis en avant un effet d'apprentissage très marqué, indiquant que le nombre de trois blocs d'essais durant cette expérience étaient sous-dimensionné. Le plan expérimental final tiendra compte de ce fort effet d'apprentissage en augmentant le nombre de blocs.

Concernant les retours utilisateurs, la majorité des remarques concernaient la technique *Screen-Space*. Tous les participants nous ont mentionné leurs difficultés à positionner l'objet en profondeur. Habitués au geste RST présent sur les produits commerciaux, les participants ont indiqué que le déplacement suivant l'axe z avec trois doigts était fastidieux. Durant cette évaluation préliminaire, notre choix des degrés de liberté contrôlés pendant le mode $2d$ se limitait aux degrés de liberté nécessaires à l'orientation : Rx, Ry et Rz. Avec deux doigts, le solveur de contraintes peut contrôler jusqu'à quatre degrés de liberté. Nous disposions donc d'un degré de liberté supplémentaire. Nous avons donc décidé de changer le mode $2d$ de la technique en incluant le degré de liberté Tz permettant le contrôle en profondeur. Ce changement est illustré dans la figure 4.2.

En résumé, cette analyse préliminaire nous a permis de construire un plan expérimental plus approprié à la tâche. Pour l'expérience finale, nous augmenterons le nombre de blocs d'essais à cinq et nous changerons le mode $2d$ de la technique *Screen-Space* en autorisant maintenant le contrôle de la profondeur (i.e. Tz) en plus du contrôle de la rotation (i.e. Rx, Ry et Rz).

4.5 Expérience contrôlée pour l'évaluation des techniques

4.5.1 Objectifs

Le but principal de cette évaluation est de déterminer l'impact de la séparation des degrés de liberté sur les écrans multipoints pour la tâche de manipulation 3-D. L'objectif secondaire est de comparer les techniques *Sticky Tools* et *Screen-Space*, jusqu'alors non évaluées ni comparées.

En considérant notre évaluation préliminaire et notre phase de création pour la technique *DS3*, nous allons tenter de vérifier avec l'étude précise les hypothèses suivantes :

En se basant à la fois sur les résultats préliminaires et sur les retours utilisateurs, nous pensons que le fait de séparer le contrôle de la rotation du contrôle de la position aura un impact positif sur les performances car les utilisateurs ne seront pas troublés, contrairement à une stratégie d'intégration liant orientation et position.

La séparation de contrôle position/orientation améliorera la coordination (à la fois en translation et rotation). En manipulant la translation et l'orientation de manière plus efficace séparément, les utilisateurs amélioreront la coordination des degrés de liberté de translation et d'orientation.

La présence de la profondeur (i.e. manipulation suivant l'axe z) durant les essais impactera à la fois performance et coordination. Cet effet sera d'autant plus marqué avec les techniques permettant le contrôle intégral de Tz et des degrés de liberté d'orientation, mettant en évidence le problème de séparation illustré pour les tâches 2-D par Nacenta et al. [79]. Malgré la volonté des utilisateurs à utiliser le geste RST, nous pensons que ce geste est moins approprié pour les tâches 3-D.

4.5.2 Tâche réalisée

La tâche ainsi que le matériel sont les mêmes que l'évaluation préliminaire, comme décrit dans les sections 4.4.1 et 4.4.2. Nous avons cependant changé l'application en ajoutant un bouton permettant aux utilisateurs de sauter un essai s'il le jugeait trop difficile à effectuer.

4.5.3 Participants

Dix hommes et deux femmes, d'une moyenne d'âge de 24.8 (écart type de 0.83) ont participé. Neuf participants étaient droitiers, trois étaient gauchers et tous possédaient une vision normale ou corrigée. Les sujets avaient une expérience variable avec les applications de réalité virtuelle et les applications 3-D mais, comme nous étudions un comportement moteur de bas niveau, cela reste acceptable. Trois participants utilisaient régulièrement un écran multipoint. Six participants étaient familiers avec la technologie tactile, comme les téléphones ou PC tactiles, mais n'avaient jamais utilisé de surfaces multipoints pendant une longue durée. Enfin, les autres participants n'avaient jamais manipulé un écran

multipoint avant l'expérience.

4.5.4 Plan expérimental

Un plan expérimental intra-sujets et une analyse à mesures répétées ont été utilisés. Les variables indépendantes étaient Technique, Profondeur, Complexité Rotation et Amplitude Rotation. Pour la variables Technique, trois niveaux étaient présents : *DS3*, *Sticky Tools* et *Screen-Space*. La présentation de cette variable était contrebalancée pour les participants.

La variable Profondeur se détaillait sur deux niveaux, suivant le besoin de déplacer l'objet le long de l'axe z pendant un essai. Nous nommerons respectivement les deux niveaux Avec Tz et Sans Tz.

Deux niveaux étaient également présents pour la variable Complexité Rotation : Simple et Complexe. Ces niveaux changeaient le type de rotation à effectuer pour correctement orienter la cheville. Dans le niveau Simple, uniquement une rotation autour d'un axe unique (x, y ou z) était nécessaire. L'axe unique était choisi de manière aléatoire. Dans le niveau Complexe, la rotation à effectuer était un mélange aléatoire des trois axes.

Enfin, la variable Amplitude Rotation possédait deux niveaux, fonction de l'amplitude totale de la rotation à effectuer. Les valeurs d'amplitude étaient respectivement 30° pour le niveau Faible et 120° dans le niveau Large.

Comme indiqué par les résultats préliminaires, nous avons augmenté le nombre de blocs à effectuer. Les participants durent donc terminer 5 blocs successifs d'essais d'expérience. Chaque bloc était composé de 16 essais : 2 répétitions de 8 combinaisons Profondeur × Complexité Rotation × Amplitude Rotation. Une pause était encouragée tous les 8 essais.

Avant de débuter l'expérience avec une nouvelle technique, les participants disposaient de 5 minutes d'entraînement pour s'exercer sur la technique. L'évaluation se terminait par un questionnaire qualitatif sur les ressentis de l'utilisateur. En moyenne, l'expérience durait 100 minutes.

Pour résumer, le plan expérimental était le suivant :
12 participants ×
3 Technique ×
5 Bloc ×
2 Profondeur ×
2 Complexité Rotation ×
2 Amplitude Rotation ×
2 répétitions
= 2,880 essais d'expérience.

4.6 Résultats et analyse

Nous orienterons la présentation des résultats autour des variables dépendantes de notre expérience. Ces variables seront subdivisées en trois catégories. Premièrement nous nous intéresserons aux indicateurs de difficulté en étudiant le nombre d'essais qui ont été

omis par les utilisateurs, ainsi que le nombre d'appuis effectués. Par la suite, nous observerons le temps de positionnement, indicateur de performance. En dernier lieu, nous examinerons plusieurs variables caractérisant la coordination ainsi que l'efficacité du mouvement. Ces indicateurs nous permettront d'analyser la perception de la tâche, de même que la stratégie utilisée par l'utilisateur. Nous utiliserons pour cela la *NDC*, la $\boldsymbol{m}$-metric, ainsi que la coordination telle que définie par Zhai [117]. Pour finir l'étude de la coordination, nous ajouterons l'étude qualitative du coefficient de translation-rotation, permettant d'évaluer la relation entre manipulation de la position et manipulation de l'orientation au cours du temps.

4.6.1 Indicateurs de difficulté

Ratio d'essais terminés

Le ratio d'essais terminés est défini comme le nombre d'essais terminés divisé par le nombre total d'essais. Les essais qui étaient omis intentionnellement par le participant (i.e. utilisation du bouton prévu à cet effet) étaient marqués comme non-terminés afin de permettre le calcul du ratio.

Une analyse à mesures répétées de la variance (ANOVA) a révélé une effet significatif de la variable TECHNIQUE ($F_{2,22} = 7.95$, $p = 0.003$). Une comparaison deux à deux a décelé une différence significative ($p \leq 0.01$) entre la technique *Screen-Space* et les autres. Le ratio d'essais terminés est de 96.0% pour *DS3*, 94.9% pour *Sticky Tools* et 86.8% pour *Screen-Space*. De prime abord, la technique *Screen-Space*, intégrant les degrés de liberté, se présente comme étant la plus difficile à utiliser pour réaliser la tâche demandée.

Nombre d'appuis

Le nombre d'appuis est défini par le nombre total d'évènements *'Appui doigt'* enregistrés durant les essais.

Une nouvelle fois, l'analyse ANOVA a trouvé un effet significatif de la variable TECHNIQUE ($F_{2,22} = 12.54$, $p = 0.002$). La comparaison deux à deux a également détecté une différence significative ($p \leq 0.02$) entre la technique *Screen-Space* et les autres. Le nombre d'appuis moyen est de 12.6 pour *DS3*, 17.2 pour *Sticky Tools* et 30.0 pour *Screen-Space*. Ce résultat est en adéquation avec le précédent. La technique *Screen-Space* nécessite de nombreux appuis pour réaliser la tâche. Cela nous indique que l'utilisateur a effectué un grand nombre d'actions séquentielles, conséquence de la difficulté rencontrée.

Cette analyse rapide nous permet d'identifier un effet concernant l'intégration des degrés de liberté. Cela semble affecter négativement l'utilisateur qui montre une plus grande difficulté à mener à bien la tâche demandée. Passons maintenant à l'analyse du temps de réalisation de la tâche afin de confirmer cette idée.

4.6.2 Temps de réalisation de la tâche

Le temps de réalisation est défini pour la tâche comme étant le temps nécessaire pour positionner avec succès la cheville dans le parallélépipède 3-D depuis le précédent

positionnement correctement réalisé. Les essais négligés par l'utilisateur sont omis dans l'analyse statistique.

Une analyse à mesures répétées de la variance (ANOVA) a révélé un effet significatif de la variable BLOC ($F_{4,44} = 4.27$, $p = 0.01$) mettant en avant la présence d'un effet d'apprentissage. Une comparaison deux à deux a décelé une différence significative ($p < 0.05$) entre le premier bloc et les autres. Par conséquent, nous avons enlevé le premier bloc pour les analyses suivantes.

L'analyse de la variance a trouvé un effet significatif de la variable TECHNIQUE sur le temps de réalisation ($F_{2,22} = 17.96$, $p < 0.001$). Une comparaison deux à deux dévoila une différence significative ($p \leq 0.001$) entre la technique *DS3* et les autres. Le temps moyen de réalisation est de 11.14s pour *DS3*, 16.81s pour *Sticky Tools* et 17.40s pour *Screen-Space*. Ce résultat supporte notre hypothèse initiale concernant la séparation des degrés de liberté. Celle-ci à un impact fort sur le temps de réalisation. La technique *DS3*, séparant les degrés de liberté de position de ceux d'orientation, est 36% plus rapide que la technique *Screen-Space* qui intègre les degrés de liberté. Une autre conclusion ressort également de ces résultats. Nous notons que la stratégie de séparation à un impact fort sur le temps de réalisation de la tâche. Les techniques *DS3* et *Sticky Tools* abordent toutes les deux des stratégies de séparation mais la technique *DS3* est 34% plus rapide que *Sticky Tools*.

Les tests statistiques ont mis en avant l'influence des variables de l'expérience. Comme prévu, l'ANOVA a détecté un effet significatif de la variable PROFONDEUR ($F_{1,11} = 23.37$, $p = 0.001$), diminuant le temps de réalisation moyen de 16.01s dans le niveau AVEC Tz à 14.14s dans le niveau SANS Tz. De même, la variable COMPLEXITÉ ROTATION a un effet significatif ($F_{1,11} = 10.37$, $p < 0.01$), diminuant le temps moyen de 16.33s dans le niveau COMPLEXE à 13.90s dans l'autre niveau. Enfin, la variable AMPLITUDE ROTATION montra également un effet significatif ($F_{1,11} = 5.98$, $p = 0.035$), diminuant le temps moyen de 17.40s dans le niveau LARGE à 12.86s dans le niveau FAIBLE. Aucune interaction entre TECHNIQUE et ces variables n'a été détectée sur le temps de réalisation.

En résumé, nous tirons de cette analyse deux conclusions concernant l'intégration des degrés de liberté pour la manipulation 3-D. D'une part, la stratégie concernant la séparation des degrés de liberté montre une influence très marquée sur la performance. Séparer le contrôle des degrés de liberté de position de ceux d'orientation apparait comme une stratégie plus performante. D'autre part, la stratégie d'intégration totale montre des performances plus faibles, résultat en accord avec les indicateurs de difficulté étudiés précédemment. Afin de comprendre et expliquer ces différence entre les stratégies, nous allons maintenant étudier en détail la coordination dans la réalisation de la tâche.

4.6.3 Coordination

Dans cette sous-section, nous étudierons la manière dont les utilisateurs ont effectué l'expérience, en nous attardant particulièrement sur la stratégie de manipulation des différents degrés de liberté. L'approche adoptée sera similaire à celle utilisée dans l'étude sur le positionnement (chapitre 3 section 3.4.2). Tout d'abord, nous analyserons les modes de la taxonomie utilisés, afin d'obtenir une vision globale. Ensuite, nous nous intéresserons au nombre de degrés de liberté manipulés simultanément. Pour évaluer cette informa-

tion, nous utiliserons la NDC. Suivant ce résultat de NDC, nous adopterons différentes stratégies. En cas de manipulation d'un unique degré de liberté (i.e. $NDC = 1$) nous identifierons précisément quel était ce degré de liberté. En cas de manipulation simultanée de multiples degrés de liberté, nous userons de la $\boldsymbol{m}$-metric pour déterminer les couples manipulés. Enfin, nous évaluerons l'efficacité quantitativement avec les indicateurs d'efficacité de Zhai, puis qualitativement avec le coefficient de translation-rotation.

Temps passé dans les différents modes

Pour commencer l'étude de la coordination, nous nous sommes intéressés au temps passé dans chacun des modes présentés dans les taxonomies, de manière à distinguer une tendance globale. Les résultats sont exposés dans le tableau 4.1.

Intitulé	*Moyenne*
Mode $1d$	38%
Mode $1d + 1i$	7%
Mode $2d$	50%
Mode $2d + 1i$	5%

(a) Technique *DS3*

Intitulé	*Moyenne*
Mode $1d$	42%
Mode $1d + 1i$	18%
Mode $2d$	34%
Mode $2d + 1i$	6%

(b) Technique *Sticky Tools*

Intitulé	*Moyenne*
Mode $1d$	35%
Mode $2d$	33%
Mode $3d$	32%

(c) Technique *Screen-Space*

TABLE 4.1 – Pourcentage de temps passé dans les différents modes.

La technique *Screen-Space* affiche des pourcentages quasi constants suivant les modes. Les utilisateurs répartissent donc leur temps équitablement entre les différents modes. Mis en relation avec la mauvaise performance sur le temps et les indicateurs de difficulté, cela nous informe que les utilisateurs ont effectué beaucoup de changements de mode, renforçant l'idée de séquentialité des actions. Concernant l'influence des variables dépendantes, deux effets intéressants ressortent de l'étude de la technique *Screen-Space*. Tout d'abord, l'ANOVA a détecté un effet de la variable PROFONDEUR uniquement pour les mode $1d$ ($F_{1,11} = 8.00$, $p = 0.03$) et $2d$ ($F_{1,11} = 15.40$, $p = 0.011$). L'absence de différence significative pour le mode $3d$ ($p = 0.55$) nous indique une tendance de l'utilisateur à manipuler le degré de liberté Tz dans le mode $2d$. Cette conclusion marque l'utilisation du geste RST pour manipuler la profondeur. Dans le niveau AVEC Tz, le temps passé dans le mode $1d$ diminue de 40 à 30% au profit du mode $2d$ qui augmente de 29 à 37%. Le second effet intéressant est celui de la variable COMPLEXITÉ ROTATION. Suivant la complexité de la rotation, l'utilisateur change marginalement le temps passé dans le mode $2d$ ($F_{1,11} = 5.65$, $p = 0.05$) et significativement le temps passé dans le mode $3d$ ($F_{1,11} = 31.78$, $p = 0.002$), sans affecter le mode $1d$ ($p = 0.17$). Dans le niveau SIMPLE, l'utilisateur consacre plus de temps au mode $2d$ (34%) qu'au mode $3d$ (26%). Dans le niveau COMPLEXE, l'inverse se produit : le mode $3d$ est plus utilisé (38%) que le mode $2d$ (29%). Cela dénote une tendance de l'utilisateur à aborder les rotations complexes avec plusieurs doigts.

L'analyse des résultats de la technique *Sticky Tools* révèle une utilisation en adéquation avec sa stratégie d'intégration des degrés de liberté. Lorsqu'un déplacement en profondeur est nécessaire, nous constatons une augmentation significative ($F_{1,11} = 5.85$, $p = 0.03$) de l'utilisation du mode $2d$, seul mode capable de manipuler le degré de liberté Tz.

L'utilisateur consacre 31% du temps dans le niveau SANS Tz et 36% dans l'autre. Cette augmentation pour le mode $2d$ est compensée par une réduction du temps passé dans les modes $1d$ et $1d+1i$. L'ANOVA a également détecté un effet de la variable COMPLEXITÉ ROTATION. En cas de rotations complexes, un changement significatif des modes $1d$ ($F_{1,11}$ = 30.71 , p < 0.001) et $2d$ ($F_{1,11}$ = 13.05 , p = 0.005) se produit tandis que les mode $1d+1i$ et $2d+1i$ ne sont pas affectés (respectivement p = 0.36 et p = 0.15). Lors de rotations complexes, l'utilisation du mode $1d$ diminue de 46% à 38% tandis que le mode $2d$ augmente de 31% à 37%. Autre résultat intéressant, le temps global passé à manipuler l'orientation (mode $1d+1i$ et $2d+1i$) est faible (24%) en comparaison avec les autres techniques. Ce faible temps, ajouté à l'invariance des modes $1d+1i$ et $2d+1i$ vis à vis de la complexité de la rotation, nous informe sur l'influence de la stratégie de séparation. Séparer les degrés de liberté de la rotation semble donc être une bonne stratégie pour la gestion de l'orientation, mais n'explique toujours pas les faibles performances réalisées avec cette technique.

Les résultats concernant la technique *DS3* sont également en adéquation avec la taxonomie. Ainsi, l'ANOVA a décelé un effet significatif de PROFONDEUR sur le temps passé dans les modes $1d+1i$ ($F_{1,11} = 45.60$, $p < 0.001$) et $2d+1i$ ($F_{1,11} = 32.98$, $p < 0.001$), seuls modes permettant le contrôle du degré de liberté Tz. Lorsque le mouvement nécessite un déplacement un profondeur, l'utilisation des modes $1d+1i$ et $2d+1i$ passe respectivement de 2% à 11% et de 3% à 7%. Même constatation pour la variable AMPLITUDE ROTATION, augmentant significativement ($F_{1,11}$ = 139.35 , p < 0.001) l'utilisation du mode $2d$ (i.e. mode consacré à la manipulation des degrés de liberté de l'orientation), passant de 41% dans le niveau FAIBLE à 59% dans l'autre. En dehors de ces résultats auxquels nous pouvions nous attendre, l'ANOVA a trouvé un résultat surprenant. La variable COMPLEXITÉ ROTATION n'a aucun effet sur le temps passé dans les modes. La complexité de la rotation ne semble donc pas affecter le temps consacré par l'utilisateur à manipuler l'orientation, conclusion similaire à la technique *Sticky Tools*.

L'étude des modes utilisés ne révèle cependant pas quels sont les degrés de liberté manipulés précisément par l'utilisateur. Afin de lever ces ambiguïtés, nous allons maintenant analyser la *NDC*, qui va nous renseigner sur l'utilisation du nombre de degrés de liberté manipulés simultanément.

Nombre de degrés de liberté manipulés

Dans un premier temps, nous allons nous intéresser au pourcentage de temps passé à manipuler un ou plusieurs degrés de liberté. Les résultats de la *NDC* sont présentés dans le tableau 4.2. Pour obtenir ces résultats nous avons seuillé les *NDC* de translation et de rotation au seuil de 0.5. (e.g. soit r le résultat, si $r = 0.3$ alors $NDC = 0$, si $r = 0.8$ alors $NDC = 1$, si $r = 1.6$ alors $NDC = 2$). Pour obtenir la *NDC* finale, nous avons sommé les *NDC* de translation et de rotation.

Nous constatons ici très clairement l'effet de l'intégration des degrés de liberté avec la technique *Screen-Space*. L'utilisateur manipule trois degrés de liberté ou plus pendant 65% du temps. Inversement, nous remarquons que les techniques *DS3* et *Sticky Tools*, favorisant la séparation des degrés de liberté, se focalisent sur la manipulation d'un ou deux degrés de liberté pendant plus de 90% de temps.

Technique	NDC1	NDC2	NDC3	NDC4	NDC5	NDC6
DS3	31%	62%	6.5%	0.5%	0%	0%
Sticky Tools	55%	33%	6.5%	5%	0.5%	0%
Screen-Space	20.5%	14%	17%	42%	6.5%	0%

TABLE 4.2 – Résultat de la *NDC* pour les différentes techniques en pourcentage du temps global.

Concernant l'influence des variables, nous nous attendions, par l'étude des modes, à trouver des effets significatifs des variables PROFONDEUR, COMPLEXITÉ ROTATION et AMPLITUDE ROTATION. Nous nous sommes donc seulement intéressés aux interactions possibles entre les techniques et ces variables, de manière à identifier des comportements différents suivant les techniques. Pour cela nous avons regroupé les valeurs de NDC3, 4, 5 et 6 en additionnant les différents pourcentages et avons effectué des analyses ANOVA. Nous réfèrerons à cet nouvel indicateur de *NDC* par la dénomination $NDC_{\geq 3}$. L'analyse ANOVA a dévoilé plusieurs interactions significatives sur $NDC_{\geq 3}$.

La première concerne une interaction TECHNIQUE × PROFONDEUR ($F_{2,22} = 6.62$, $p = 0.02$). Lorsque l'utilisateur a besoin de déplacer l'objet en profondeur, la comparaison deux à deux détecta deux effets significatifs. Le premier effet concerne la technique *Sticky Tools*. Dans le niveau AVEC Tz, l'utilisateur change sa stratégie de manipulation et intègre plus de degrés de liberté, comportement illustré par une augmentation significative ($p < 0.01$) de la $NDC_{\geq 3}$ passant de 10% à 16%. Cela indique potentiellement un effet involontaire de la technique, similaire au problème exposé par Nacenta et al [79]. En effet, lorsque l'utilisateur souhaite déplacer l'objet en profondeur, il se trouve dans le mode $2d$ ou $2d + 1i$. Dans ces modes, il contrôle de manière intégrale Tz et Rz et peut donc provoquer un changement d'orientation involontaire lors d'un déplacement en profondeur. Le second effet est propre à la technique *DS3*. Cette fois-ci, le passage au niveau AVEC Tz entraine une diminution significatif ($p = 0.012$) de la $NDC_{\geq 3}$. Cette diminution illustre l'utilisation du mode $1d+1i$ où le degré de liberté Tz est manipulé séparément. Ces résultats nous donnent une première piste sur l'explication de la différence de performance entre les techniques *Sticky Tools* et *DS3* : séparer les degrés de liberté de position et d'orientation permettrait d'éviter des mouvements involontaires, mouvements qui diminuent la performance.

La seconde interaction trouvée par l'ANOVA concerne TECHNIQUE × COMPLEXITÉ ROTATION ($F_{2,22} = 11.75$, $p < 0.004$). Une comparaison deux à deux a détecté une différence significative uniquement présente pour les techniques *Sticky Tools* ($p = 0.03$) et *Screen-Space* ($p = 0.006$). La technique *DS3* ne semble pas affectée ($p = 0.92$). Une rotation complexe entraine une hausse de la $NDC_{\geq 3}$ de 10% à 16% pour *Sticky Tools* et de 61% à 70% pour *Screen-Space*. Nous obtenons une nouvelle piste sur l'explication de la différence de performance entre les techniques, la technique *DS3* contrôlant un nombre de degrés de liberté réduits en comparaison avec les autres.

Grâce à la *NDC*, nous avons identifié des pistes probables sur l'explication des différences entre les techniques. Nous pensons ainsi qu'en fournissant un contrôle séparé des degrés de liberté de position et d'orientation, les utilisateurs manipulent plus fréquemment

un nombre réduit de degrés de liberté et effectuent moins de mouvements involontaires. Ces différences expliqueraient la meilleure performance de la technique *DS3*. Afin de valider ces hypothèses, nous allons dans un premier temps examiner les degrés de liberté manipulés séparément.

Identification des degrés de liberté manipulés séparément

Afin de procéder à l'identification des degrés de liberté manipulés séparément, nous avons utilisé le résultat de la NDC1. Nous avons cherché si cette valeur provenait de la *NDC* de translation (respectivement d'orientation) et nous avons extrait le degré de liberté de position (respectivement d'orientation) présentant le plus grand changement, puis calculé un ratio. Les résultats sont exposés dans le tableau 4.3

Technique	Tx	Ty	Tz	Rx	Ry	Rz	Total (NDC1)
DS3	8.5%	15.5%	1%	3.5%	1%	1.5%	31%
Sticky Tools	9.5%	20.5%	0%	10.5%	13.5%	1%	55%
Screen-Space	6.5%	13%	0%	0.5%	0.5%	0%	20.5%

TABLE 4.3 – Pourcentage de temps passé à manipuler un degré de liberté unique.

La première information qui sort de ces résultats est le taux relativement important de temps passé à manipuler séparément les degrés de liberté Tx et Ty. Pour toutes les techniques, la manipulation de Tx et Ty monopolise plus de la moitié du temps consacré à la manipulation de degré de liberté unique.

Concernant la technique *DS3*, nous remarquons que le degré de liberté Tz est manipulé seulement 1% du temps séparément. Dans la taxonomie, ce degré de liberté est contrôlé de manière séparée dans les modes $1d + 1i$ et $2d + 1i$. Ce résultat illustre une capacité de l'utilisateur à paralléliser ses actions, en utilisant des doigts différents. Le faible pourcentage de temps passé à manipuler les degrés de liberté de rotation est également consistant avec leur contrôle intégral dans la technique.

Pour la technique *Screen-Space*, nous constatons un comportement en adéquation avec la taxonomie. L'utilisateur a réussi à séparer uniquement les degrés de liberté du mode $1d$, degrés contrôlées par un doigt unique. Pour les autres modes, le résultat proche de 0% nous indique une grande difficulté à effectuer des manipulations précises.

Les résultats spécifiques à la technique *Sticky Tools* nous révèlent deux comportements intéressants. Premièrement, les degrés de liberté Rx et Ry, contrôlés séparément dans les modes $1d + 1i$ et $2d + 1i$, sont manipulés dans le même ordre de grandeur de temps. Contrairement à la technique *DS3* où l'utilisateur semble avoir majoritairement parallélisé les actions du doigt indirect, nous remarquons ici une stratégie de manipulation orientée autour de la séparation. Cela nous fournit une conclusion intéressante : l'utilisateur serait plus enclin à paralléliser ses actions pour contrôler un degré de liberté unique (i.e. Tz dans la technique *DS3*) plutôt que plusieurs degrés de liberté (i.e. Rx et Ry dans la technique *Sticky Tools*). Ensuite, nous remarquons que les degrés de liberté Tz et Rz, contrôlés intégralement dans le mode $2d$, présentent un très faible taux d'utilisation séparée. Cela pourrait indiquer des mouvements volontaires induits par le geste RST sur les degrés de liberté intégrés.

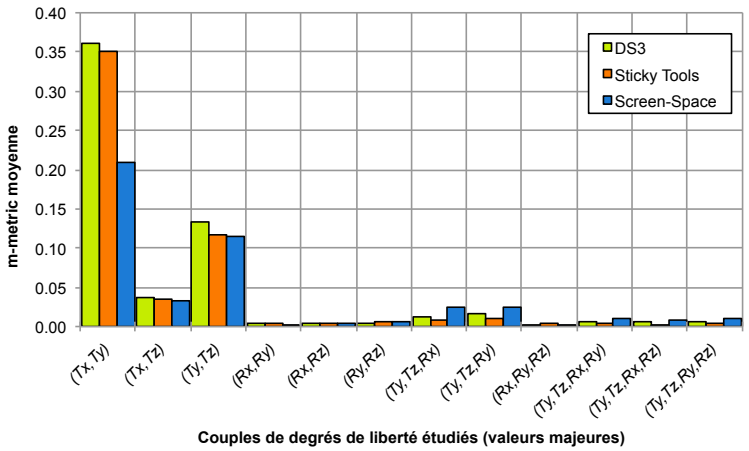


FIGURE 4.5 – Valeurs moyennes pour la $\boldsymbol{m}$-metric et les différents couples étudiés. Nous constatons la très forte domination du couple (Tx, Ty).

Pour vérifier l'hypothèse concernant ces mouvements involontaires, passons maintenant à l'identification et à l'analyse des degrés de liberté manipulés intégralement.

Identification des degrés de liberté manipulés intégralement

Dans cette sous-section, nous allons tenter de préciser les hypothèses effectuées précédemment. Pour cela, nous allons évaluer la $\boldsymbol{m}$-metric puis relier ces résultats aux modes des taxonomies. En comparaison au temps passé dans les modes, la $\boldsymbol{m}$-metric nous fournit en plus une indication d'efficacité car seuls les mouvements de l'utilisateur rapprochant (i.e. en position ou orientation) la cheville de la destination sont pris en compte. Les moyennes sont exposées dans la figure 4.5.

Nous avons été très surpris des résultats de la $\boldsymbol{m}$-metric. En effet, ceux-ci sont très faibles (la $\boldsymbol{m}$-metric est normalisée entre 0 et 1) en comparaison avec ceux calculés dans le chapitre sur le positionnement. Cette métrique, illustrant à la fois l'efficacité (i.e. uniquement les réductions d'erreurs sont prises en compte) et la simultanéité, informe de la tendance générale à effectuer une manipulation coordonnée pour atteindre le but. La seule conclusion que nous tirons de l'analyse de la $\boldsymbol{m}$-metric est que seul le couple (Tx, Ty) est utilisé de manière coordonnée et efficacement, c'est à dire en réduisant l'erreur simultanément en position pour l'axe x et y. Pour les autres couples étudiés, les faibles scores nous informent d'une efficacité réduite.

Pour cependant analyser quels étaient les degrés de liberté manipulés intégralement, nous avons adopté une démarche similaire à celle expliquée dans la section précédente. Quand nous trouvions une valeur de *NDC* égale à deux, nous avons cherché le couple de degrés de liberté manipulés intégralement puis établi une moyenne d'utilisation représentée en pourcentage de temps. La même procédure à été adoptée pour les autres valeurs de la *NDC*. Dans un soucis de clarté, nous présenterons uniquement les résultats pertinents à chaque technique, permettant de comprendre les stratégies utilisateurs. L'ensemble de ces résultats est visible sur le tableau 4.4.

Commençons par la technique *Screen-Space*. Les résultats des principaux couples ma-

Couple	Temps
(Ty, Tz, Ry, Rz)	14.5%
(Ty, Tz, Rx, Rz)	14.5%
(Tx, Tz)	9.5%
(Ty, Tz, Rx, Ry)	6.5%
(Ty, Tz, Rx)	6%
(Ty, Tz, Ry)	6%
Autres	$\leq 2\%$

(a) Technique *Screen-Space*

Couple	Temps
(Tx, Tz)	11.5%
(Ty, Tz)	6%
(Rx, Rz)	5.5%
(Ry, Rz)	4%
(Rx, Ry)	3.5%
(Ty, Tz, Ry, Rz)	2.5%
Autres	$\leq 2\%$

(b) Technique *Sticky Tools*

Couple	Temps
(Ty, Tz)	20.5%
(Rx, Rz)	13%
(Ry, Rz)	13%
(Tx, Tz)	9.5%
(Rx, Ry, Rz)	6.5%
(Rx, Ry)	6%
Autres	$\leq 2\%$

(c) Technique *DS3*

TABLE 4.4 – Pourcentage de temps passé à manipuler intégralement des couples de degrés de liberté.

nipulés intégralement sont exposés dans la table 4.4a. Nous retrouvons dans ces résultats la stratégie d'intégration propre à la technique. Si nous combinons ces résultats avec la faible performance, les indicateurs de difficulté et le faible score de la $\boldsymbol{m}$-metric cela indique une difficulté de la part de l'utilisateur à effectuer des actions précises. Notre hypothèse de départ est vérifiée : en manipulant intégralement plusieurs degrés de liberté l'utilisateur effectue ainsi de nombreuses modifications de position et d'orientation involontaires, modifications qui donnent lieu à des rectifications expliquant la faible performance ainsi le grand nombre d'appuis. Pour la manipulation 3-D, l'approche intégrale est donc à éviter.

Concernant la technique *Sticky Tools*, les résultats nous dévoilent deux comportements intéressants. D'une part, nous constatons que les deux couples les plus utilisés contiennent le degré de liberté Tz. Ajouté au faible temps d'utilisation séparée de ce degré de liberté, ce résultat nous informe sur l'utilisation du mode $2d$ de la technique ainsi que la manipulation de Tz. L'utilisateur éprouve des difficultés à manipuler indépendamment Tz, ce qui engendre des changements involontaires d'autres degrés de liberté ici illustrés par Tx et Ty. D'autre part, ces résultats nous révèlent que les degrés de liberté Rx et Ry sont rarement contrôlés de manière intégrale.

Pour la technique *DS3*, les résultats confortent la précédente conclusion sur l'utilisation parallèle du doigt direct et indirect. Les couples (Tx, Tz) et (Ty, Tz) sont utilisés pendant 30% du temps et illustrent des actions combinées et parallèles de la part de l'utilisateur. Celui-ci réussit donc à intégrer, à l'aide de l'interaction bi-manuelle, des degrés de liberté initialement séparés. Les forts taux d'utilisation incluant les degrés de liberté de rotation sont en adéquation avec leur contrôle intégral. Un autre résultat intéressant est présent pour *DS3*. Les utilisateurs manipulent plus souvent le couple (Ty, Tz) que le couple (Tx, Tz). Nous pensons que ce résultat provient de la stratégie bimanuelle employée. En effet, il est plus aisé pour l'utilisateur d'effectuer avec chaque main un mouvement de va-et-vient (manipulant Ty avec une main et Tz avec l'autre) que des mouvements de natures différentes (mouvement de gauche à droite pour Tx et de va-et-vient pour Tz).

Ces conclusions, combinées avec celle concernant la manipulation séparée, nous informent de l'importance de la stratégie de séparation. Avec la technique *DS3*, l'utilisateur réussit à paralléliser ses actions, contrôlant ainsi le degré de liberté séparé Tz

simultanément avec les degrés de liberté Tx et Ty. Avec la technique *Sticky Tools*, l'utilisateur ne parallélise pas ses actions, et contrôle les degrés de liberté séparés (i.e. Rx et Ry) indépendamment. Nous pensons que cela provient de la structure de manipulation qui s'offre à chaque doigt. Dans la technique *DS3*, le doigt indirect, ainsi que les doigts direct, contrôlent un attribut de position. Cela permet d'appliquer les préconisations de Guiard [41] sur la manipulation à deux mains. Guiard introduit la notion de chaine cinématique reliant l'usage des mains dominantes et non-dominantes. Largement approuvé par la communauté, son modèle indique que la main non-dominante est chargée de modifier le contexte pour la main dominante, capable ainsi de coordonner des actions simultanées. En manipulation la position avec l'aide des deux mains, la technique *DS3* est compatible avec ce modèle. La technique *Sticky Tools* quant à elle, mélange des attributs d'orientation avec des attributs de position, rendant le processus cognitif difficile à paralléliser pour l'utilisateur. Cette conclusion, en complément de l'hypothèse des mouvements involontaires, explique la différence de performances entre les techniques utilisant des stratégies de séparation.

Nous venons de nous consacrer à l'identification précise des degrés de liberté manipulés intégralement et séparément. Nous avons identifié deux causes expliquant la différence de performance : la parallélisation d'actions par des doigts contrôlant des attributs de natures différentes, ainsi que l'intégration de degrés de liberté de natures différentes. Pour valider ces causes, nous allons maintenant étudier l'efficacité telle que décrite par Zhai [117]. Pour cela nous allons dans un premier temps nous attarder sur l'efficacité de translation. Ensuite nous adopterons une approche similaire pour l'efficacité de rotation. Ces résultats ne prenant pas en compte l'influence du déplacement sur l'orientation, nous finirons par l'étude qualitative du coefficient de translation-rotation.

Efficacité de translation

L'efficacité de translation est définie comme le ratio de la longueur du chemin le plus court par la longueur du chemin effectivement parcouru du début (i.e. cheville positionnée aléatoirement) à la fin (i.e. cheville correctement positionné) d'un essai d'expérience [116].

L'analyse ANOVA révéla un effet significatif de la variable Technique ($F_{2,22} = 112.19$, $p < 0.001$). Une comparaison deux à deux montra une différence significative ($p < 0.001$) entre la technique *Screen-Space* et les autres. La comparaison deux à deux dévoila également une différence marginale ($p = 0.06$) entre les techniques *DS3* et *Sticky Tools*. L'efficacité moyenne est de 0.63 pour *DS3*, 0.56 pour *Sticky Tools* et enfin 0.26 pour *Screen-Space*. À nouveau, la technique intégrant tous les degrés de liberté termine en dernière position. En séparant le contrôle de position, la technique *DS3* possède donc la meilleur efficacité en translation.

Les tests statistiques ont également montré l'influence de la variable Profondeur. L'ANOVA identifia une différence significative ($F_{1,11} = 1286.73$, $p < 0.001$) et une interaction significative Technique × Profondeur ($F_{2,22} = 19.81$, $p < 0.001$). Dans le niveau Sans Tz, la technique *DS3* devance *Sticky Tools* ($p = 0.01$) et *Screen-Space* ($p < 0.001$). Pour ce niveau, l'efficacité de translation moyenne est de 0.81 pour la technique *DS3*, 0.69 pour *Sticky Tools* et 0.34 pour *Screen-Space* (figure 4.6). La différence significative entre les techniques *Sticky Tools* et *DS3* n'est pas présente dans l'autre niveau. Cela

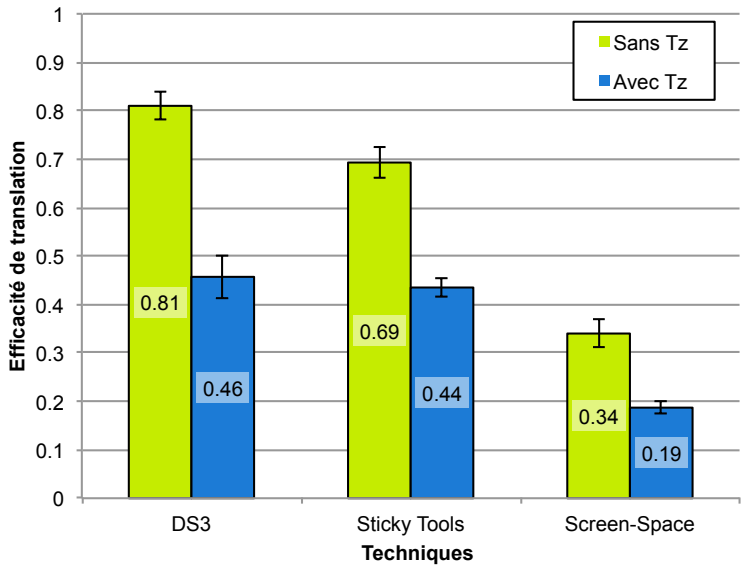


FIGURE 4.6 – Efficacité de translation moyenne pour chaque technique dans les différents niveaux de PROFONDEUR. Les barres d'erreur représentent l'intervalle de confiance à 95%.

renforce une nouvelle fois l'hypothèse des mouvements involontaires. Lorsque l'utilisateur n'a pas besoin d'effectuer de déplacement en profondeur, il effectue involontairement des modifications de Tz avec les techniques intégrant ce degré de liberté, résultant en une efficacité de translation plus faible. Nous pensons que les mouvements involontaires sont également responsables de l'effet significatif de la variable TECHNIQUE.

L'analyse ANOVA a également révélé un effet significatif de la variable AMPLITUDE ROTATION ($F_{1,11} = 208.25$, $p < 0.001$). Le niveau LARGE diminue l'efficacité de translation de 0.56 à 0.41. Plus intéressant, nous avons trouvé une interaction significative TECHNIQUE × AMPLITUDE ROTATION ($F_{2,22} = 17.32$, $p = 0.001$). Dans le niveau FAIBLE, la technique *DS3* devance significativement *Sticky Tools* ($p < 0.01$) et *Screen-Space* ($p = 0.001$). Pour ce niveau, l'efficacité de translation moyenne est de 0.69 pour la technique *DS3*, 0.60 pour *Sticky Tools* et 0.40 pour *Screen-Space*. Ce résultat conforte et précise l'hypothèse des mouvements involontaires. Lorsque de faibles rotations sont nécessaires, les techniques manipulant simultanément des attributs de position et d'orientation effectuent des changements de rotation qui affecte la position, nécessitant par la suite un autre changement, responsable de la diminution d'efficacité en translation. Ceci nous indique que les mouvements involontaires sont produits lors de manipulations précises.

De manière plus évidente, l'ANOVA trouva aussi un effet significatif de la variable COMPLEXITÉ ROTATION ($F_{1,11} = 16.49$, $p < 0.010$). Les rotations complexes diminuent l'efficacité de translation moyenne de 0.52 à 0.46.

Enfin, l'analyse à mesures répétées a trouvé deux interactions significatives : AMPLITUDE ROTATION × PROFONDEUR ($F_{1,11} = 23.42$, $p = 0.005$) et TECHNIQUE × AMPLITUDE ROTATION × PROFONDEUR ($F_{2,22} = 8.93$, $p = 0.006$). Dans le niveau de AMPLITUDE ROTATION FAIBLE et le niveau de PROFONDEUR SANS Tz , la technique *DS3* devance *Sticky Tools* ($p < 0.001$) et *Screen-Space* ($p = 0.002$). Pour ces niveaux, l'efficacité de translation moyenne est de 0.87 pour la technique *DS3*, 0.74 pour *Sticky*

Tools et 0.55 pour *Screen-Space*. Une nouvelle fois, ce résultat supporte l'hypothèse des mouvements involontaires, les techniques manipulation translation et rotation rendent difficile l'exécution d'actions précises sur un degré de liberté unique.

Efficacité de rotation

L'efficacité de rotation est définie comme le ratio du décalage en amplitude de rotation initiale par l'amplitude de rotation effectuée durant un essai d'expérience [116].

L'analyse ANOVA a trouvé un effet significatif de la variable TECHNIQUE ($F_{2,22} = 11.71$, $p < 0.005$). Une comparaison deux à deux décela une différence significative entre les techniques *Sticky Tools* et *Screen-Space* ($p = 0.02$), et entre *DS3* et *Screen-Space* ($p = 0.02$). L'efficacité de rotation moyenne est de 0.48 pour la technique *Sticky Tools*, 0.39 pour *DS3* et 0.28 pour *Screen-Space*. La stratégie d'intégration des degrés de liberté se révèle être encore une fois la moins efficace. Le bon résultat de la technique *Sticky Tools* nous indique un autre effet de la stratégie de séparation adoptée par les techniques. Contrairement à la technique *DS3* où les degrés de liberté de rotation sont intégrés, *Sticky Tools* offre une manipulation séparée des degrés de liberté de rotation et montre la meilleure efficacité de rotation. Au delà de la séparation position/orientation, ce résultat nous indique qu'une séparation plus approfondie, spécifique à la rotation, produit de bons résultats. Mise en relation avec le temps passé à effectuer le contrôle d'orientation (section 4.6.3), cela nous donne une piste d'amélioration pour une nouvelle technique qui intégrerait les stratégies de séparation propres à *DS3* et à *Sticky Tools*. Ce résultat reste néanmoins en suspend car une meilleure efficacité selon Zhai n'implique pas forcément de meilleures performances.

Une nouvelle fois, l'ANOVA a trouvé un effet significatif de la variable PROFONDEUR ($F_{1,11} = 12.57$, $p < 0.005$). L'efficacité de rotation moyenne est de 0.48 dans le niveau SANS Tz et de 0.39 dans le niveau AVEC Tz. Plus intéressant, l'ANOVA a trouvé une interaction significative TECHNIQUE × PROFONDEUR ($F_{2,22} = 11.87$, $p < 0.005$) (figure 4.7). Une comparaison deux à deux révéla que les techniques *Sticky Tools* et *Screen-Space* réduisaient leur efficacité de rotation moyenne ($p < 0.01$) en changeant le niveau de variable PROFONDEUR de SANS Tz à AVEC Tz. La technique *DS3* elle, n'affiche pas de différence significative ($p = 0.6$). En passant au mode AVEC Tz, les techniques ont diminué leur efficacité de rotation moyenne respectivement de 0.53 à 0.44 pour *Sticky Tools* et de 0.33 à 0.23 pour *Screen-Space*. Une nouvelle fois, cela confirme l'hypothèse des mouvements involontaires, affectant cette fois-ci l'efficacité de rotation. En manipulant Tz, les techniques intégrant ce degré de liberté avec les degrés de liberté d'orientation effectuent des rotations involontaires, provoquant une diminution de l'efficacité de rotation.

De plus, l'ANOVA a trouvé un effet significatif de la variable COMPLEXITÉ ROTATION ($F_{1,11} = 33.17$, $p < 0.002$), les rotations complexes diminuant l'efficacité de rotation moyenne de 0.44 à 0.33. Plus intéressant, nous avons trouvé une interaction significative TECHNIQUE × COMPLEXITÉ ROTATION ($F_{2,22} = 17.97$, $p < 0.001$). La comparaison deux à deux a identifié que l'augmentation de la complexité de rotation (i.e. passage du niveau SIMPLE à COMPLEXE) réduit l'efficacité de rotation moyenne uniquement pour les techniques *Sticky Tools* et *Screen-Space* ($p < 0.001$). Une nouvelle fois, *DS3* n'est pas affectée significativement ($p = 0.68$). En passant au niveau COMPLEXE, l'efficacité

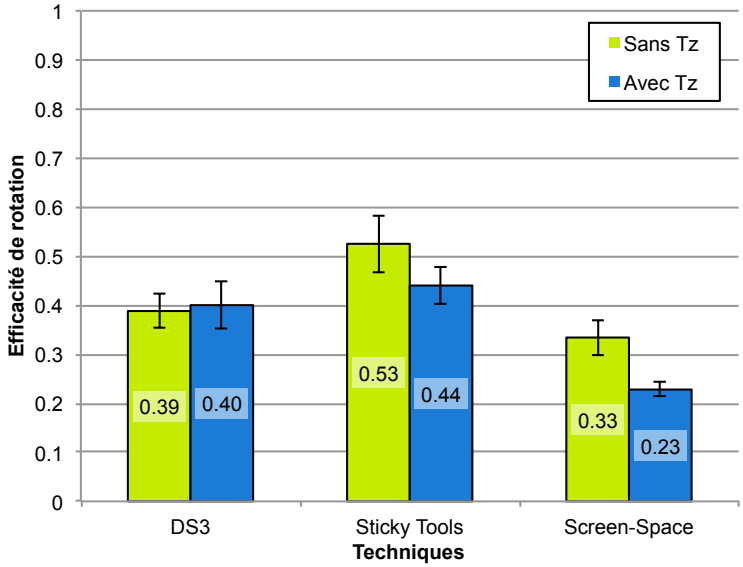


FIGURE 4.7 – Efficacité de rotation moyenne pour chaque technique dans les différents niveaux de PROFONDEUR. Les barres d'erreur représentent l'intervalle de confiance à 95%.

de rotation moyenne diminue de 0.58 à 0.38 pour *Sticky Tools* et de 0.34 à 0.22 pour *Screen-Space*. Ce résultat relativise notre précédente conclusion sur une nouvelle technique mixant les bienfaits de *Sticky Tools* et *DS3*. En effet, l'absence de différence significative entre les techniques *Sticky Tools* et *DS3* dans le niveau COMPLEXE met en avant le bon comportement de la stratégie d'intégration des degrés de liberté de rotation.

Finalement, l'ANOVA a trouvé deux interactions significatives concernant les variables PROFONDEUR et AMPLITUDE ROTATION : PROFONDEUR × AMPLITUDE ROTATION ($F_{1,11} = 37.91$, $p = 0.002$) et TECHNIQUE × PROFONDEUR × AMPLITUDE ROTATION ($F_{2,22} = 7.82$, $p < 0.01$). La comparaison deux à deux révéla un effet n'ayant lieu que dans le niveau FAIBLE de la variable AMPLITUDE ROTATION. Dans ce niveau là, changer la variable PROFONDEUR au niveau AVEC Tz entraine une baisse significative de l'efficacité de rotation uniquement pour les techniques *Sticky Tools* ($p < 0.01$) et *Screen-Space* ($p = 0.001$). Une nouvelle fois, la technique *DS3* n'est pas affectée par le changement de la variable PROFONDEUR ($p = 0.96$). Dans le niveau AVEC Tz, l'efficacité de rotation diminue de 0.53 à 0.40 pour la technique *Sticky Tools* et de 0.41 à 0.19 pour *Screen-Space*. Une nouvelle fois, cela illustre le comportement stable d'une gestion intégrée des degrés de liberté de rotation.

Relation entre efficacité de translation et efficacité de rotation

Les deux précédentes mesures d'efficacité permettent d'évaluer respectivement le déplacement effectué en position et en rotation. Cependant, ces mesures ne tiennent pas compte de l'influence d'un déplacement sur l'autre. Par exemple, l'utilisateur peut très bien se rapprocher du but en position tout en affectant négativement l'orientation. Pour évaluer cette influence, Zhai a introduit un ratio, évaluant en fonction du temps l'efficacité en translation et l'efficacité en rotation. Ce ratio se présente sous la forme d'une courbe

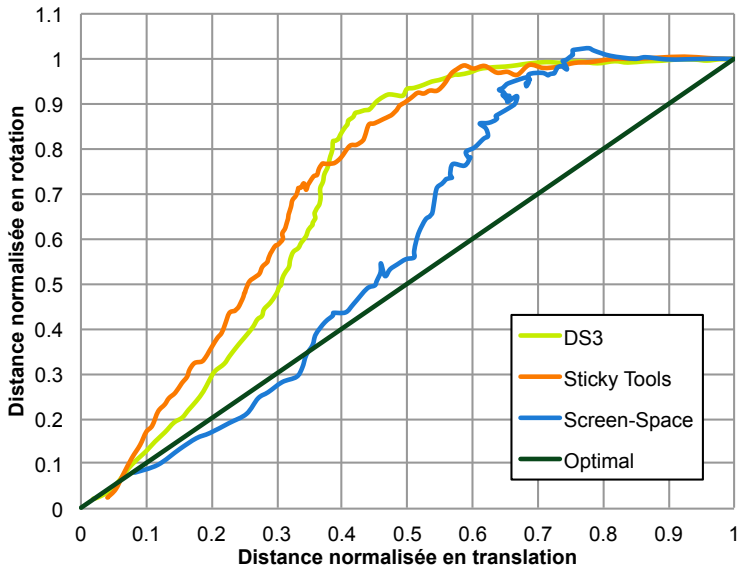


FIGURE 4.8 – Ratio de translation-rotation moyen pour chaque technique. Les courbes débutent aux coordonnées $(1, 1)$ et terminent aux coordonnées $(0, 0)$. Le paramètre temps n'est pas représenté.

paramétrée, affichée dans un espace 2-D présentant en abscisse l'efficacité de translation et en ordonné l'efficacité en rotation. Nous avons choisi d'échantillonner la courbe sur 100 points. La courbe débute au point de coordonnées $(1, 1)$ (i.e. à $t = 0$ l'objet n'est ni positionné, ni orienté) et termine au point de coordonnée $(0, 0)$ (i.e. objet correctement positionnée). Les courbes des trois techniques sont exposées dans la figure 4.8.

Pour les trois techniques, nous constatons que l'utilisateur commence par réduire l'efficacité de translation avant de débuter la rotation. L'allure générale de la courbe est similaire pour les techniques *DS3* et *Sticky Tools* et nous pouvons distinguer deux phases. L'utilisateur commence par manipuler majoritairement les degrés de liberté de position. Il réduit ainsi l'efficacité de translation de 0.6 (i.e. de 1 à 0.4) tout en diminuant l'efficacité de rotation de 0.2 (i.e. passage de 1 à 0.8). Ensuite, il manipule majoritairement la rotation, diminuant ainsi l'efficacité de rotation de 0.8 (i.e. de 0.8 à 0) tout en manipulant la position de 0.4 (i.e. de 0.4 à 0).

Concernant la technique *Screen-Space*, le profil est différent. La rotation commence plus tôt en comparaison avec les autres techniques. De façon générale, la courbe moyenne de la technique *Screen-Space* est plus proche de l'optimal que les courbes moyennes des autres techniques. Ce résultat est à mettre en perspective avec les faibles performances de la techniques : même si la stratégie d'intégration permet une manipulation simultanée de la position et de l'orientation qui améliore le ratio d'efficacité, cette stratégie est à éviter pour la manipulation 3-D sur écran multipoint. Une meilleure coordination n'implique donc pas de meilleures performances.

Nous voulions également mettre en avant les mouvements involontaires produits par les utilisateurs pour les techniques manipulant simultanément les degrés de liberté de position et d'orientation. Les courbes moyennes compensant les écarts entre les utilisateurs, nous avons choisi d'illustrer ceci en affichant uniquement la courbe moyenne propre à

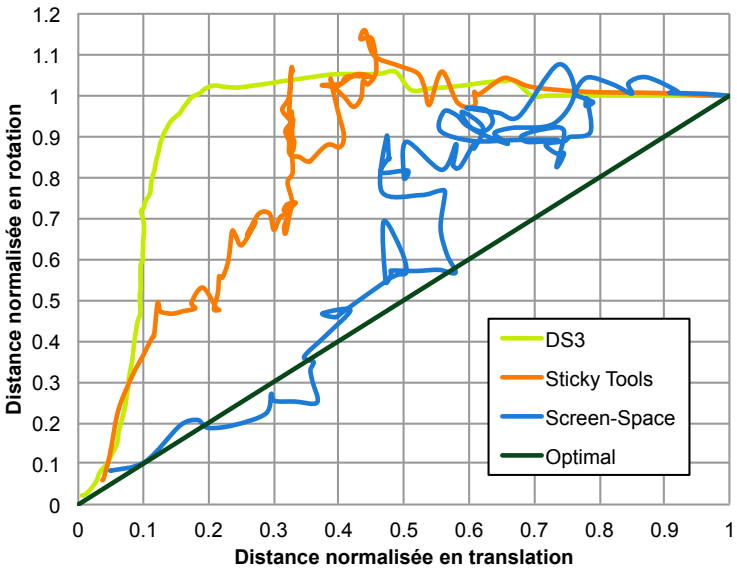


FIGURE 4.9 – Exemples de courbes typiques de différents participants pour chaque technique. Illustration du contrôle simultanée avec les techniques *Sticky Tools* et *Screen-Space* et de la séparation avec la technique *DS3*.

un utilisateur, illustrant clairement notre point de vue. Ces courbes se trouvent sur la figure 4.9 et illustrent particulièrement bien les mouvements involontaires effectués dans les techniques *Screen-Space* et *Sticky Tools*.

En résumé, l'analyse de la coordination nous a fournit des explications sur les performances des techniques. Tout d'abord la stratégie d'intégration de la technique *Screen-Space* oblige l'utilisateur à de multiples actions successives, du fait des mouvements involontaires induits par l'intégration des degrés de liberté. Ensuite la différence entre les stratégies de séparation des techniques *Sticky Tools* et *DS3* s'explique par l'influence de deux facteurs. Premièrement la gestion des degrés de liberté contrôlés séparément par un doigt indirect. Nous avons conclu qu'il fallait que les degrés de liberté soient de même nature afin de permettre une parallélisation des actions. Deuxièmement l'intégration de degrés de liberté de différentes natures. En séparant clairement le contrôle de position du contrôle d'orientation, la technique *DS3* est ainsi la plus performante. Nous allons maintenant passer au retour qualitatif fourni par les utilisateurs à la suite de l'expérience.

4.6.4 Analyse qualitative

Parmi les 12 participants, 8 ont préféré la technique *DS3*, 2 ont noté à égalité les techniques *DS3* et *Sticky Tools* et enfin 2 ont choisi *Sticky Tools*. Les participants ayant désigné *DS3* ont trouvé la technique plus facile d'utilisation, et ont apprécié le fait que la rotation soit découplée de la translation. Selon eux, cela permettait un meilleur contrôle de leurs actions. Ils ont également mentionné le fait que la rotation était facile à effectuer et *naturelle* (i.e. naturelle dans le sens où les actions du participant résultaient en un changement d'orientation voulu par celui-ci). Les sujets qui ont préféré la technique *Sticky Tools* trouvèrent la technique plus adaptée pour une manipulation à deux mains. Nos

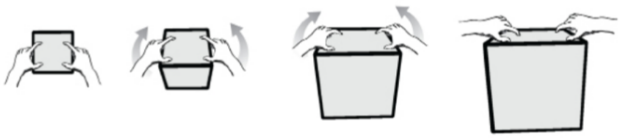


FIGURE 4.10 – Illustration du geste à 4 doigts avec la technique *Screen-Space* de Reisman et al. [88].

conclusions vont cependant dans le sens inverse : même si le contrôle bimanuel est possible avec *Sticky Tools* il empêche cependant une manipulation parallèle. Certains participants nous ont également indiqué qu'ils étaient capables d'effectuer les actions requises avec une seul main et que cette possibilité était appréciable. Contrairement aux autres utilisateurs, ils n'apprécièrent pas la séquentialité offerte par *DS3* concernant la manipulation séparé de l'orientation de la position.

Spécifiquement à propos de *Sticky Tools*, un utilisateur a mentionné que la manière d'effectuer les rotations n'était ni efficace, ni intuitive. De plus, il n'a pas aimé le couplage entre la translation en profondeur (i.e. manipulation Tz) et la rotation (i.e. manipulation Rz). Cette difficulté induite par ce couplage translation/rotation fut également reporté par 5 autres participants. Cela confirme notre hypothèse concernant les mouvements aléatoires et la supériorité de la stratégie de séparation position/orientation. Un autre participant nous à néanmoins mentionné que contrôler la rotation par le doigt indirect était plus facile que contrôler la position en profondeur (i.e. dans la technique *DS3*).

Concernant la technique *Screen-Space*, tous les participants nous ont fait part de leurs difficultés à utiliser la technique pour réaliser la tâche. Deux participants nous ont signalé que, même si la technique semblait intuitive au premier abord, il était très difficile de faire des actions précises. Quatre participants ont apprécié la technique lors de la manipulation d'une surface plane. Ils ont utilisé avec succès un geste introduit par Reisman et al. (voir figure 4.10), où quatre doigts permettent la manipulation d'un plan 3-D. Cependant, ils nous indiquèrent que la technique était limitée de part la taille des objets. Plus l'objet était loin de la caméra, plus il apparaissait petit à l'écran et plus il devenait difficile d'utiliser la technique *Screen-Space* avec succès. Un autre participant nous a également indiqué que les mouvements complexes étaient faciles à obtenir avec la technique tandis que les mouvements simples (i.e. rotation autour d'un seul axe, translation en profondeur) étaient difficiles à réaliser. Ce retour fut également mentionné par deux autres participants qui qualifièrent la technique d'*imprévisible*. Cette remarque renforce la précédente idée sur la réalisation d'actions précises, difficiles à réaliser en cas d'intégration des degrés de liberté. Cela explique le nombre important d'appuis réalisés, conséquences des nombreuses actions à réaliser pour atteindre l'objectif.

4.7 Discussion et conclusion

Nous avons conçu l'expérience dans le but de comparer pour la première fois trois techniques de manipulation 3-D sur écrans multipoints. Les trois techniques diffèrent

dans leur manière de manipuler les degrés de liberté nécessaires. La technique *Screen-Space* contrôle les six degrés de liberté nécessaire intégralement alors que les techniques *DS3* et *Sticky Tools* séparent les degrés de liberté en utilisant deux stratégies différentes.

4.7.1 Séparation des degrés de liberté et performance

Les résultats nous montrent que, pour les techniques étudiées, intégrer les degrés de liberté de la tâche de manipulation amène une diminution des performances. La technique *Screen-Space* est ainsi 53% plus lente que la technique *DS3* qui adopte une stratégie de séparation.

Ce résultat fut expliqué par l'étude de la coordination, à la fois en translation et rotation, illustrant l'efficacité à réaliser la tâche. La technique *Screen-Space*, qui lie ensemble de manière forte translation et rotation, s'est révélée avoir la plus mauvaise coordination (i.e. en translation et rotation). La technique *Sticky Tools* améliore la coordination en translation de 115% et la coordination en rotation de 71% par rapport à *Screen-Space*. La technique *DS3* quant à elle, améliore la coordination de translation par 142% et la coordination de rotation par 39%. Nous avons également montré l'existence de mouvements involontaires induits par cette intégration des degrés de liberté, expliquant le grand nombre d'actions à réaliser successivement et expliquant la faible performance.

Il apparait donc que pour la tâche intégrale de manipulation 3-D considérée, l'utilisation d'une stratégie d'intégration totale des degrés de liberté entraine une performance moindre. Au contraire, il apparait que suivre la structure du périphérique d'entrée (i.e. séparée pour un écran multipoint), amène à de meilleurs résultats. Cette conclusion va dans le sens du travail effectué par Veit et al. [100], qui ont trouvé des résultats similaires pour des tâches de rotation en réalité virtuelle.

La conclusion des travaux de Jacob [55] sur l'intégration des degrés de liberté était de faire correspondre le plus possible la structure de la tâche avec la structure du périphérique d'entrée. Pour un écran multipoint, suivre de telles préconisations est impossible pour la manipulation 3-D. Notre étude suggère que pour optimiser la performance, il convient de suivre la structure du périphérique d'entrée plutôt que celle de la tâche.

4.7.2 Stratégies de séparation des degrés de liberté

L'analyse statistique a trouvé une différence significative pour le temps de complétion entre les techniques *DS3* et *Sticky Tools*, avec une amélioration de 34% pour *DS3*. Cela nous montre que la stratégie de séparation des degrés de liberté peut avoir un impact fort sur les performances.

La définition de Garner [40] nous a poussé à caractériser la manipulation 3-D comme une tâche à deux attributs intégraux, la position et l'orientation, rendant l'application des théories de Jacob difficile lors de l'utilisation d'écrans multipoints. Le temps de complétion plus rapide pour la technique *DS3* suggère que la position et l'orientation sont malgré tout deux attributs différents que l'utilisateur peut facilement séparer.

Lors de la création de la technique *DS3*, nous avons choisi de séparer clairement le contrôle des degrés de liberté de translation de ceux de rotation, améliorant non seulement le temps mais également la coordination en translation. En contrôlant la position

en profondeur en même temps que l'orientation, *Sticky Tools* produit une coordination de translation plus faible lorsque la tâche ne nécessite aucun changement de profondeur. Cela illustre une difficulté de la part des utilisateurs à séparer le contrôle de l'orientation avec la technique *Sticky Tools*. Nous avons illustré cette idée dans la figure 4.9 qui montre clairement les mouvements involontaires produits par cette intégration des degrés de liberté.

Ces conclusions étendent à la manipulation 3-D les travaux de Nacenta et al. [79]. Avec des techniques liant position et orientation, il devient plus difficile d'effectuer de larges rotations sans affecter la position de l'objet, résultant par une faible performance et une faible efficacité. En adoptant une stratégie de séparation claire pour la technique *DS3*, nous avons amélioré les performances, l'efficacité et la satisfaction utilisateur.

4.7.3 Stratégie d'interaction bimanuelle

La stratégie de séparation n'est cependant pas seule responsable de l'écart de performance entre les techniques. Nous avons trouvé une autre conséquence, relatif à l'utilisation bi-manuelle. En effet, l'utilisation d'écrans multipoints offre la possibilité d'utiliser plusieurs doigts mais également plusieurs mains. Pour la technique *Sticky Tools*, une main peut ainsi contrôler la position de l'objet et l'autre main peut effectuer des rotations de manière indirecte. L'étude de l'intégration des degrés de liberté nous permet de dire que pour permettre un contrôle parallèle des deux mains, il convient de faire correspondre la nature des degrés de liberté contrôlés par chaque main. La technique *Sticky Tools* permet de manipuler des degrés de liberté de position avec une main et des degrés de liberté d'orientation avec l'autre main. Comme exposé dans les résultats, cette différence de type d'attributs ne permet pas un contrôle simultané des deux mains. De part la charge cognitive induite, l'utilisateur ne sait pas coordonner correctement ses deux mains. La technique *DS3* contrôle quant à elle le même type d'attributs avec les deux mains, les degrés de liberté de position. Cela permet à l'utilisateur de paralléliser ses actions, participant ainsi à une amélioration des performances.

4.7.4 Contrôle direct vs. contrôle indirect

Un des points clé de la technique *DS3* est le changement de mode pour passer du contrôle de la position au contrôle d'orientation. Dans notre cas, nous effectuons le changement de mode en fonction du nombre de doigt en contact. Cependant, cela peut avoir un gros défaut quand il s'agit de manipuler des petits objets. Par exemple, utiliser trois doigts avec la technique *Screen-Space* s'est avéré très difficile pour des petits objets. Le contrôle indirect représente une façon de résoudre ce problème.

Un contrôle indirect offre la possibilité de séparer clairement les degrés de liberté, tout en permettant également la manipulation de petits objets. Néanmoins, il est possible avec le contrôle indirect, de manipuler involontairement des objets proches. Cette limitation est cependant aisément contournable avec une zone d'interaction indirecte dédiée, apparaissant autour du premier objet sélectionnée. Dans sa forme simple, ce rendu pourrait être un cercle semi-transparent apparaissant autour de l'objet. En plus de résoudre le problème lié aux petits objets, cette zone pourra également permettre le contrôle de plusieurs

objets. Cette faculté est aujourd'hui présente uniquement avec la technique *Screen-Space*. Les techniques *DS3* et *Sticky Tools* n'autorisent pas cette fonction.

Pour résumé, nous avons comparé trois techniques d'interaction permettant la manipulation 3-D sur écrans multipoints. Nous avons démontré la supériorité des techniques d'interaction favorisant la séparation à l'intégration des degrés de liberté de libertés. Nous avons ensuite analysé les raisons de cette supériorité pour mettre en avant les stratégies de séparation efficaces, permettant un contrôle simultané avec les deux mains. Ces préconisations d'ordre général pourront servir de base aux créateurs d'interfaces 3-D pour dispositifs multipoints. Notons également que la technique *DS3* a été utilisée lors d'une exposition nommée *Numériquement Vôtre* à Cap Sciences [15] pour l'atelier Hypnotus, atelier consacré à la présentation de la réalité augmentée et de la manipulation 3-D.

15. `http://tinyurl.com/numeriquement-votre`

Conclusion

Notre travail visait à étudier le lien entre les périphériques d'entrées multipoints et la tâche de manipulation 3-D, plus particulièrement l'influence de la stratégie de séparation des degrés de liberté.

Bilan global

Nous avons débuté par un état de l'art traitant du sujet. Nous avons ainsi caractérisé la technologie multipoint, présenté les théories concernant la séparation des degrés de liberté et listé les techniques d'interaction existantes. Nous avons enfin exposé différentes métriques permettant de caractériser la coordination lors de la réalisation de tâches de manipulation 3-D. Nous avons remarqué la présence de différentes stratégies d'intégration pour les techniques de manipulation 3-D, différences qui jusqu'alors n'avaient pas été abordées.

Dans un second chapitre, nous avons présenté les taxonomies existantes pour représenter la tâche de manipulation 3-D. Puis, nous avons explicité leurs limites, principalement liées à la généricité offerte par leurs conceptions. En se basant sur cette analyse, nous avons introduit une nouvelle taxonomie, propre aux écrans multipoints et à la manipulation 3-D. Cette taxonomie nous à permis tout au long du mémoire de visualiser les différences de conception quant aux stratégies d'intégration des degrés de liberté pour les différentes techniques de manipulation 3-D.

Nous nous sommes ensuite consacrés à une sous-tâche de la manipulation 3-D, à savoir le positionnement 3-D. Nous avons introduit deux techniques mettant à profit les possibilités offertes par les écrans multipoints. La première, intitulée *Z-technique*, permet une interaction plein-écran tandis que la seconde, intitulée *MT-viewport*, sépare l'écran en quatre zones. Ces deux techniques ont été évaluées et nous avons démontré qu'elles offraient des performances équivalentes. D'un point de vue coordination, nous avons mis en avant l'influence de la précision sur la stratégie suivie l'utilisateur. Plus la tâche devient difficile (i.e plus la tâche nécessite une grande précision), plus l'utilisateur ajuste séquentiellement les degrés de liberté de positionnement. De plus, notre étude a mis en avant l'influence du contrôle indirect offert par la *Z-technique*. Avec cette technique, l'utilisateur parvient à intégrer plus facilement les degrés de liberté, indiquant une action simultanée de ses deux mains. Pour l'autre technique, présentant le même objet dans différentes zones de l'écran, nous n'avons pas retrouvé cette simultanéité dans les actions, illustrant la charge cognitive plus importante induite par les multiples représentations.

Enfin, nous avons consacré un chapitre à l'étude de la manipulation 3-D. En se basant sur les techniques existantes, nous avons introduit une nouvelle technique, *DS3*, qui re-

prend la même stratégie de positionnement que la *Z-technique* tout en intégrant ensemble les degrés de liberté nécessaires à l'orientation. L'évaluation de cette nouvelle technique avec les techniques déjà existantes nous révéla que cette nouvelle technique était plus rapide pour effectuer la tâche de manipulation 3-D. Nous nous sommes également penchés sur la perception ainsi que la réalisation de la tâche par l'utilisateur. L'étude avancée de la coordination pour les différentes techniques nous dévoila de nombreux résultats. Tout d'abord, nous savons maintenant qu'une stratégie d'intégration globale des degrés de liberté résulte par des performances amoindries. Plus précisément, nous avons décelé qu'une stratégie de séparation des degrés de liberté de position (respectivement d'orientation) offrait une coordination en translation (respectivement en rotation) plus importante dans la réalisation de la tâche. Ensuite, nous avons mis en avant l'importance de la conception des techniques de manipulation 3-D vis-à-vis de l'utilisation de deux mains. Nous préconisons ainsi d'offrir la manipulation d'attributs de même type par les deux mains afin de pouvoir paralléliser leurs mouvements.

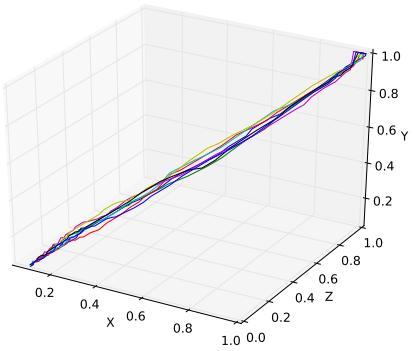
Perspectives de recherche

Les différents résultats obtenus dans le cadre de ce mémoire donnent naissance à plusieurs pistes de recherches :

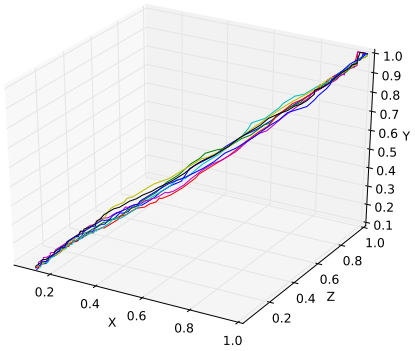
- L'étude de la manipulation 3-D s'est conclue avec la préconisation d'une nouvelle technique, basée sur une séparation des degrés de liberté de position mais également d'orientation. Toujours dans le même cadre d'étude, il serait intéressant de comparer cette nouvelle technique avec la technique *DS3*.
- Concernant le comportement moteur de l'utilisateur, nous avons constaté une forte influence de la précision nécessaire pour réaliser une tâche de manipulation 3-D. Essayer de déterminer une adaptation des techniques d'interaction fonction de la précision voulue semble être une piste intéressante. Nous pouvons imaginer une technique intégrant les degrés de liberté pour effectuer des actions rapides et grossières, qui s'adapterait ensuite pour effectuer des mouvements précis en séparant automatiquement les degrés de liberté. Des travaux récent commencent d'ailleurs à aborder cette problématique [101].
- Dans tous nos travaux, nous avons effectué des études utilisant une caméra virtuelle fixe. Les tâches de navigation 3-D, où nous ajustons manuellement la caméra virtuelle est en ce sens une perspective de recherche pouvant mettre à profit nos résultats.
- Plus généralement, les travaux effectués dans ce mémoire préconisent des stratégies pour la conception de techniques d'interactions qui doivent ensuite être adaptées et appliquées pour la réalisation d'applications concrètes. La suite logique serait donc de travailler main dans la main avec un utilisateur professionnel de logiciels 3-D , que ce soit pour la conception ou l'assemblage assisté par ordinateur (CAO). Lors de l'usage de ce type de logiciel, de nombreuses actions de sélections et de manipulation successives sont nécessaires et l'utilisation de tel outils peut ainsi s'avérer laborieuse. Dans cette optique, la manipulation 3-D à l'aide d'écrans multipoints pourraient éventuellement simplifier l'usage de ces logiciels. Une étude précise en collaboration avec un professionnel serait la première étape nécessaire pour cette approche.

Annexe A

Trajectoires des techniques de positionnement 3-D

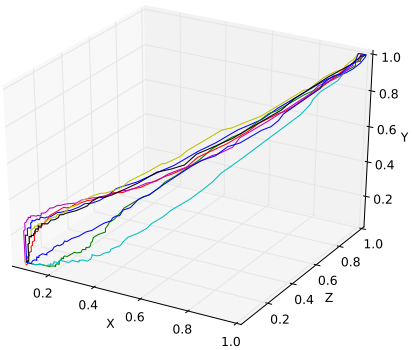


(a) Trajectoire 3-D avec besoin de précision.

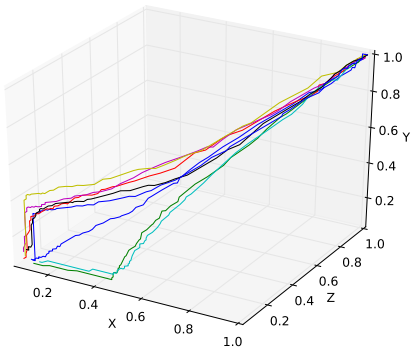


(b) Trajectoire 3-D sans besoin de précision.

FIGURE A.1 – Courbe de trajectoire 3-D pour la technique *Z-technique*.

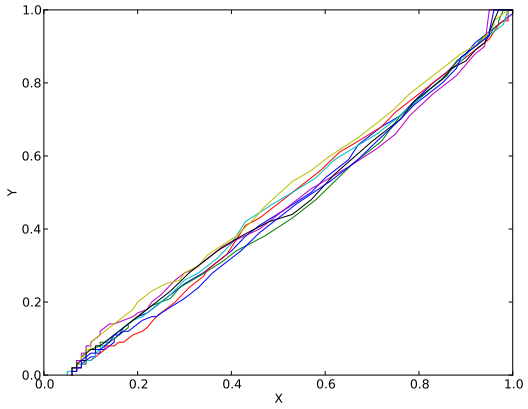


(a) Trajectoire 3-D avec besoin de précision.

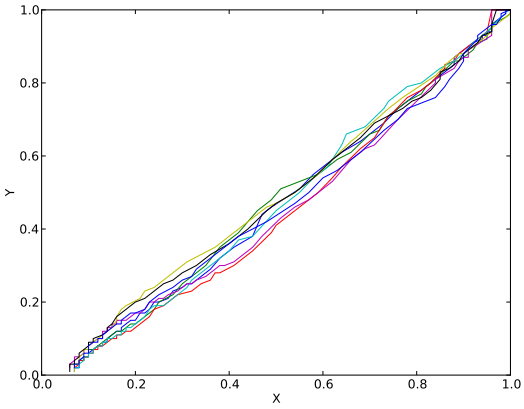


(b) Trajectoire 3-D sans besoin de précision.

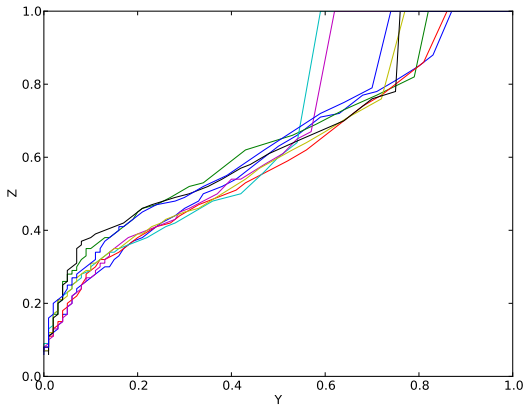
FIGURE A.2 – Courbe de trajectoire 3-D pour la technique *MT-viewport*.



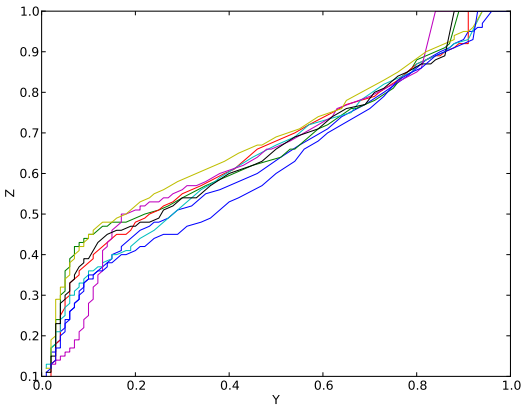
(a) Sur le plan XY avec besoin de précision.



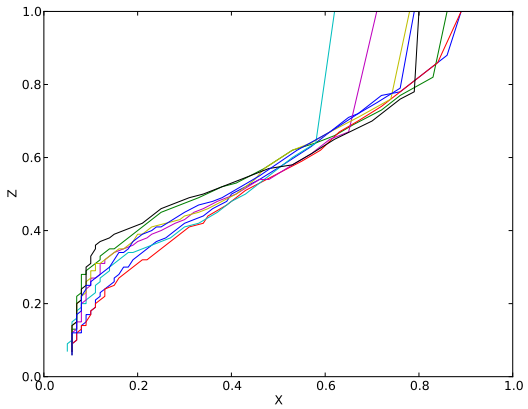
(b) Sur le plan XY sans besoin de précision.



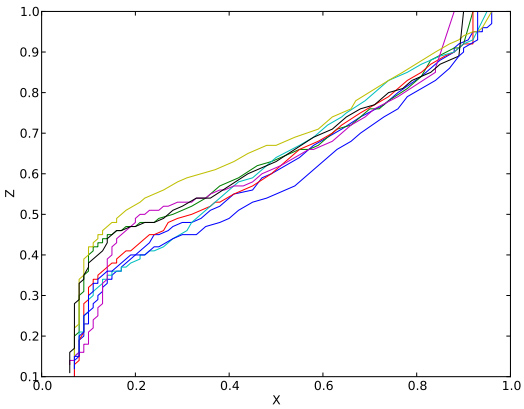
(c) Sur le plan YZ avec besoin de précision.



(d) Sur le plan YZ sans besoin de précision.

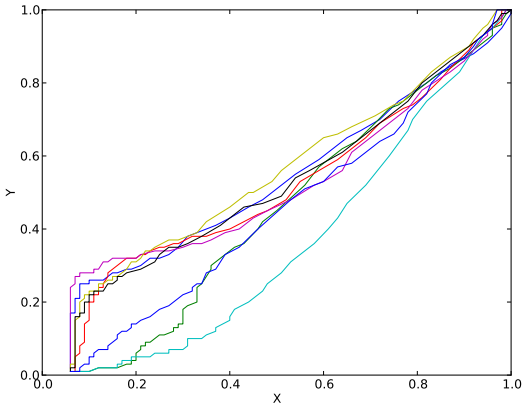


(e) Sur le plan XZ avec besoin de précision.

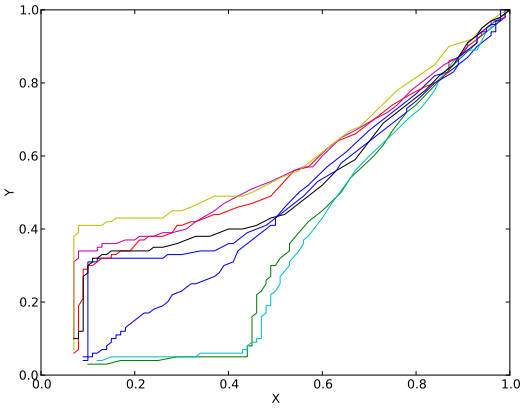


(f) Sur le plan XZ sans besoin de précision.

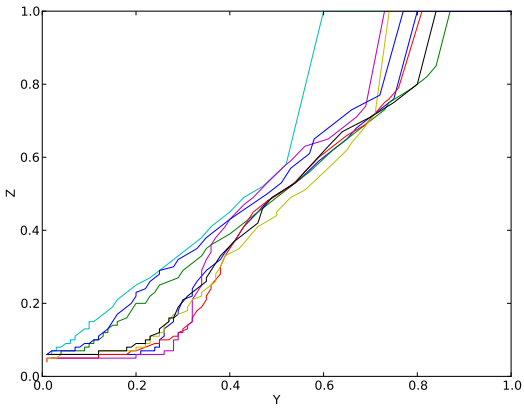
FIGURE A.3 – Courbe de trajectoire 2-D pour la technique *Z-technique*.



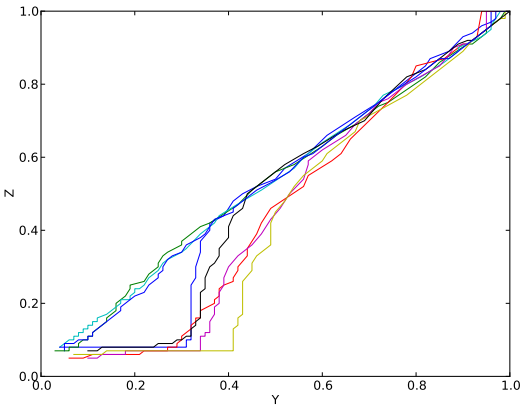
(a) Sur le plan XY avec besoin de précision.



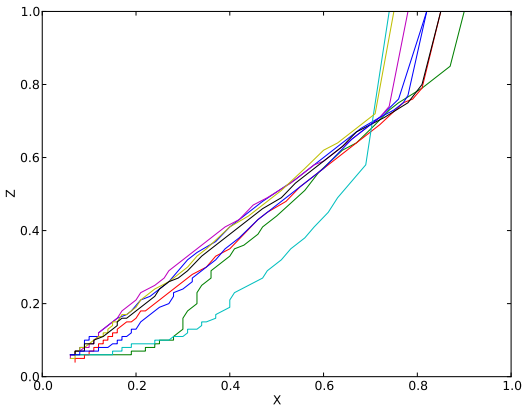
(b) Sur le plan XY sans besoin de précision.



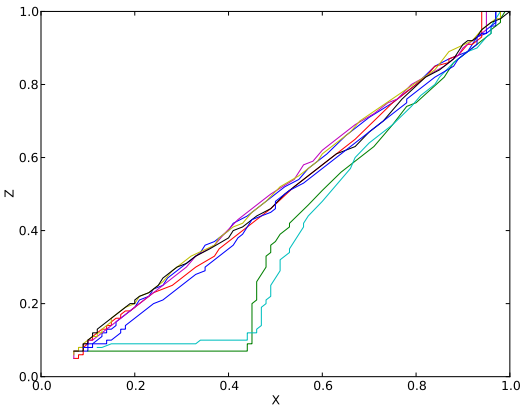
(c) Sur le plan YZ avec besoin de précision.



(d) Sur le plan YZ sans besoin de précision.



(e) Sur le plan XZ avec besoin de précision.



(f) Sur le plan XZ sans besoin de précision.

FIGURE A.4 – Courbe de trajectoire 2-D pour la technique *MT-viewport*.

Bibliographie

[1] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 105–112, Ft. Lauderdale, Florida, USA, 2003. ACM.

[2] Michel Amberg, Frédéric Giraud, Betty Semail, Paolo Olivo, Géry Casiez, and Nicolas Roussel. Stimtac : a tactile input device with programmable friction. In *Proceedings of the 24th annual ACM symposium adjunct on User interface software and technology*, UIST '11 Adjunct, pages 7–8, New York, NY, USA, 2011. ACM.

[3] Fred Attneave. Dimensions of similarity. *American Journal of Psychology*, 63(4) :516–556, 1950.

[4] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. The Rockin'Mouse : integral 3D manipulation on a plane. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 311–318, Atlanta, Georgia, United States, 1997. ACM.

[5] Ravin Balakrishnan and Ken Hinckley. Symmetric bimanual interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 33–40, The Hague, The Netherlands, 2000. ACM.

[6] H. Benko and S. Feiner. Balloon selection : A Multi-Finger technique for accurate Low-Fatigue 3D selection. In *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, 2007.

[7] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, page 1263–1272, Montréal, Québec, Canada, 2006. ACM.

[8] François Bérard and Yann Laurillau. Single user multitouch on the diamondtouch : from 2 x 1d to 2d. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 1–8, New York, NY, USA, 2009. ACM.

[9] Eric A. Bier. Snap-dragging in three dimensions. In *Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 193–204, Snowbird, Utah, United States, 1990. ACM.

[10] Eric A. Bier and Maureen C. Stone. Snap-dragging. *SIGGRAPH Comput. Graph.*, 20(4) :233–240, 1986.

[11] Eric Allan Bier. Skitters and jacks : interactive 3D positioning tools. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 183–196, Chapel Hill, North Carolina, United States, 1987. ACM.

[12] M. Biet, G. Casiez, F. Giraud, and B. Lemaire-Semail. Discrimination of virtual square gratings by dynamic touch on friction based tactile displays. In *Proceedings of the 2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, HAPTICS '08, pages 41–48, Washington, DC, USA, 2008. IEEE Computer Society.

[13] Richard A. Bolt. Put-that-there : Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '80, pages 262–270, New York, NY, USA, 1980. ACM.

[14] Richard A. Bolt. Put-that-there : Voice and gesture at the graphics interface. *SIGGRAPH Comput. Graph.*, 14 :262–270, July 1980.

[15] Doug A. Bowman and Larry F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. pages 35–38, 1997.

[16] Doug A. Bowman, Donald B. Johnson, and Larry F. Hodges. Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 26–33, London, United Kingdom, 1999. ACM.

[17] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces : Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[18] Richard W. Bukowski and Carlo H. Séquin. Object associations : a simple and practical approach to virtual 3D manipulation. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 131–ff., Monterey, California, United States, 1995. ACM.

[19] W. Buxton and B. Myers. A study in two-handed input. *SIGCHI Bull.*, 17(4) :321–326, 1986.

[20] William Buxton. Lexical and pragmatic considerations of input structures. *SIGGRAPH Comput. Graph.*, 17(1) :31–37, 1983.

[21] William Buxton. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction*, page 449–456. North-Holland Publishing Co., 1990.

[22] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. The design space of input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Empowering people*, pages 117–124, Seattle, Washington, United States, 1990. ACM.

[23] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.*, 9(2) :99–122, 1991.

[24] G. Casiez and C. Chaillou. Effects of dof separation on elastic devices for the navigation in 3d virtual environments with force feedback. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 483 – 486, march 2005.

[25] G. Casiez, P. Plénacoste, C. Chaillou, and B. Semail. The digihaptic, a new three degrees of freedom multi-finger haptic device. *Proceedings of Virtual Reality International Conference*, pages 35–39, 2003.

[26] Géry Casiez. *Contribution à l'étude des interfaces haptiques. Le DigiHaptic : un périphérique haptique de bureau à degrés de liberté séparés.* PhD thesis, Université des Sciences et Technologies de Lille, 2004.

[27] Géry Casiez and Nicolas Roussel. No more bricolage ! : methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 603–614, New York, NY, USA, 2011. ACM.

[28] Géry Casiez, Nicolas Roussel, Romuald Vanbelleghem, and Frédéric Giraud. Efficacité et robustesse aux distracteurs d'un retour tactile pour faciliter le pointage. In *Conference Internationale Francophone sur I'Interaction Homme-Machine*, IHM '10, pages 25–32, New York, NY, USA, 2010. ACM.

[29] Géry Casiez, Nicolas Roussel, Romuald Vanbelleghem, and Frédéric Giraud. Surfpad : riding towards targets on a squeeze film effect. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2491–2500, New York, NY, USA, 2011. ACM.

[30] Géry Casiez and Daniel Vogel. The effect of spring stiffness and control gain with an elastic rate control pointing device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1709–1718, New York, NY, USA, 2008. ACM.

[31] Géry Casiez, Daniel Vogel, Qing Pan, and Christophe Chaillou. Rubberedge : reducing clutching by combining position and rate control with elastic feedback. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 129–138, New York, NY, USA, 2007. ACM.

[32] Géry Casiez, Patricia Plénacoste, and Christophe Chaillou. Does dof separation on elastic devices improve user 3d steering task performance ? In Masood Masoodian, Steve Jones, and Bill Rogers, editors, *Computer Human Interaction*, volume 3101 of *Lecture Notes in Computer Science*, pages 70–80. Springer Berlin Heidelberg, 2004.

[33] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human–Computer Interaction*, 23(3) :215–250, 2008.

[34] Michael Chen, S. Joy Mountford, and Abigail Sellen. A study in interactive 3-D rotation using 2-D control devices. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 121–129. ACM, 1988.

[35] Aurélie Cohé, Fabrice Dècle, and Martin Hachet. tbox : a 3d transformation widget designed for touch-screens. In *Proceedings of the 2011 annual conference on Human*

factors in computing systems, CHI '11, pages 3005–3008, New York, NY, USA, 2011. ACM.

[36] Brookshire D. Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 183–188, Cambridge, Massachusetts, United States, 1992. ACM.

[37] Paul Dietz and Darren Leigh. DiamondTouch : a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, page 219–226, Orlando, Florida, 2001. ACM.

[38] Donato Fiorella, Andrea Sanna, and Fabrizio Lamberti. Multi-touch user interface evaluation for 3d object manipulation on mobile devices. *Journal on Multimodal User Interfaces*, 4 :3–10, 2010. 10.1007/s12193-009-0034-4.

[39] James D. Foley, Victor L. Wallace, and Peggy Chan. The human factors of computer graphics interaction techniques. *IEEE Comput. Graph. Appl.*, 4(11) :13–48, 1984.

[40] W.R. Garner. *The Processing of Information and Structure*. 1974.

[41] Yves Guiard. *Asymmetric Division of Labor in Human Skilled Bimanual Action : The Kinematic Chain as a Model*, volume 19. 1987.

[42] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, page 115–118, Seattle, WA, USA, 2005. ACM.

[43] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. Shallow-depth 3d interaction : design and evaluation of one-, two- and three-touch techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 1147–1156, San Jose, California, USA, 2007. ACM.

[44] Mark Hancock, Thomas T. Cate, and Sheelagh Carpendale. Sticky tools : Full 6dof force-based interaction for multi-touch tables. In *Proceedings of Interactive Tabletops and Surfaces 2009*, 2009.

[45] Mark S. Hancock, Sheelagh Carpendale, Frédéric D. Vernier, Daniel Wigdor, and Chia Shen. Rotation and translation mechanisms for tabletop interaction. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, page 79–88. IEEE Computer Society, 2006.

[46] Andrew J. Hanson. The rolling ball. In *Graphics Gems III*, pages 51–60. Academic Press, 1992.

[47] Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the air : adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 139–148, Victoria, BC, Canada, 2009. ACM.

[48] 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, 1997.

[49] Uta Hinrichs and Sheelagh Carpendale. Gestures in the wild : studying multi-touch gesture sequences on interactive tabletop exhibits. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 3023–3032, New York, NY, USA, 2011. ACM.

[50] Steve Hodges, Shahram Izadi, Alex Butler, Alban Rrustemi, and Bill Buxton. ThinSight : versatile multi-touch sensing for thin form-factor displays. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, page 259–268, Newport, Rhode Island, USA, 2007. ACM.

[51] Jeff Hultquist. Graphics gems. chapter A virtual trackball, pages 462–463. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[52] Immersion. Immersion. http ://www.immersion.fr/, 2009.

[53] James N Ingram, Konrad P Körding, Ian S Howard, and Daniel M Wolpert. The statistics of natural hand movements. *Experimental Brain Research. Experimentelle Hirnforschung. Expérimentation Cérébrale*, 188(2) :223–236, June 2008. PMID : 18369608.

[54] Robert J. K. Jacob and Linda E. Sibert. The perceptual structure of multidimensional input device selection. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 211–218, Monterey, California, United States, 1992. ACM.

[55] Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and Jr M. Preston Mullen. Integrality and separability of input devices. *ACM Trans. Comput.-Hum. Interact.*, 1(1) :3–26, 1994.

[56] R H Jacoby, M Ferneau, and J Humphries. Gestural interaction in a virtual environment. *Proceedings of SPIE*, 2177(May 2011) :355, 1994.

[57] Kenrick Kin, Tom Miller, Björn Bollensdorff, Tony DeRose, Björn Hartmann, and Maneesh Agrawala. Eden : a professional multitouch tool for constructing virtual organic environments. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 1343–1352, New York, NY, USA, 2011. ACM.

[58] Sebastian Knoedel and Martin Hachet. Multi-touch rst in 2d and 3d spaces : Studying the impact of directness on user performance. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, pages 75 –78, march 2011.

[59] Myron W. Krueger. *Artificial reality.* Addison-Wesley, 1983.

[60] Myron W. Krueger. *Artificial reality II.* Addison-Wesley, 1991.

[61] Myron W. Krueger. Full-body interactive exhibits. In *ICHIM*, pages 222–234, 1991.

[62] Myron W. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. Videoplace–an artificial reality. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '85, pages 35–40, New York, NY, USA, 1985. ACM.

[63] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. Fluid integration of rotation and translation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 601–610, Portland, Oregon, USA, 2005. ACM.

[64] Celine Latulipe, Craig S. Kaplan, and Charles L. A. Clarke. Bimanual and unimanual image alignment : an evaluation of mouse-based techniques. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, page 123–131, Seattle, WA, USA, 2005. ACM.

[65] S. K. Lee, William Buxton, and K. C. Smith. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 21–25, San Francisco, California, United States, 1985. ACM.

[66] Jun Liu, David Pinelle, Samer Sallam, Sriram Subramanian, and Carl Gutwin. TNT : improved rotation and translation on digital tables. In *Proceedings of Graphics Interface 2006*, pages 25–32, Quebec, Canada, 2006. Canadian Information Processing Society.

[67] Jock Mackinlay, Stuart K. Card, and George G. Robertson. A semantic analysis of the design space of input devices. *Hum.-Comput. Interact.*, 5(2) :145–190, 1990.

[68] Sylvain Malacria, Eric Lecolinet, and Yves Guiard. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces : the cyclostar approach. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 2615–2624, New York, NY, USA, 2010. ACM.

[69] A. Martinet, G. Casiez, and L. Grisoni. The design and evaluation of 3d positioning techniques for multi-touch displays. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 115 –118, march 2010.

[70] A. Martinet, G. Casiez, and L. Grisoni. Integrality and separability of multitouch interaction techniques in 3d manipulation tasks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(3) :369 –380, march 2012.

[71] Anthony Martinet, Géry Casiez, and Laurent Grisoni. 3d positioning techniques for multi-touch displays. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, VRST '09, pages 227–228, New York, NY, USA, 2009. ACM.

[72] Anthony Martinet, Géry Casiez, and Laurent Grisoni. The effect of dof separation in 3d manipulation tasks with multi-touch displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, pages 111–118, New York, NY, USA, 2010. ACM.

[73] Anthony Martinet, Laurent Grisoni, and Géry Casiez. Comparing most recent 3d manipulation techniques for multi-touch displays. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, pages 310–310, New York, NY, USA, 2010. ACM.

[74] Maurice R. Masliah and Paul Milgram. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 25–32, The Hague, The Netherlands, 2000. ACM.

[75] A H Mason and P J Bryden. Coordination and concurrency in bimanual rotation tasks when moving away from and toward the body. *Experimental Brain Research.*

Experimentelle Hirnforschung. Expérimentation Cérébrale, 183(4) :541–556, December 2007. PMID : 17668195.

[76] Nimish Mehta. A Flexible Machine Interface. Master's thesis, Department of Electrical Engineering, University of Toronto, 1982.

[77] Mark Mine. Virtual environment interaction techniques. Technical report, UNC Chapel Hill CS Dept, 1995.

[78] Tomer Moscovich and John F. Hughes. Multi-finger cursor techniques. In *Proceedings of Graphics Interface 2006*, page 1–7, Quebec, Canada, 2006. Canadian Information Processing Society.

[79] Miguel A. Nacenta, Patrick Baudisch, Hrvoje Benko, and Andy Wilson. Separability of spatial manipulations in multi-touch interfaces. In *Proceedings of Graphics Interface 2009*, pages 175–182, Kelowna, British Columbia, Canada, 2009. Canadian Information Processing Society.

[80] Gregory M. Nielson and Jr Dan R. Olsen. Direct manipulation techniques for 3D objects using 2D locator devices. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 175–182, Chapel Hill, North Carolina, United States, 1987. ACM.

[81] OGRE. Open source 3D graphics engine. http ://www.ogre3d.org/.

[82] Ji-Young Oh and Wolfgang Stuerzlinger. Moving objects with 2D input devices in CAD systems and desktop virtual environments. In *Proceedings of Graphics Interface 2005*, pages 195–202, Victoria, British Columbia, 2005. Canadian Human-Computer Communications Society.

[83] Alex Olwal, Steven Feiner, and Susanna Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 295–304, Florence, Italy, 2008. ACM.

[84] Mark Platshon. Acoustic touch technology adds a new input dimension. *Computer Design*, pages 89–93, 1988.

[85] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa. Egocentric object manipulation in virtual environments : Empirical evaluation of interaction techniques. 1998.

[86] 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 9th annual ACM symposium on User interface software and technology*, UIST '96, pages 79–80, New York, NY, USA, 1996. ACM.

[87] Ivan Poupyrev and Tadao Ichikawa. Manipulating objects in virtual worlds : Categorization and empirical evaluation of interaction techniques. *Journal of Visual Languages and Computing*, 10 :19–35, 1999.

[88] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. A screen-space formulation for 2D and 3D direct manipulation. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 69–78, Victoria, BC, Canada, 2009. ACM.

[89] Jun Rekimoto. SmartSkin : an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems : Changing our world, changing ourselves*, page 113–120, Minneapolis, Minnesota, USA, 2002. ACM.

[90] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. TapTap and MagStick : improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces*, page 146–153, Napoli, Italy, 2008. ACM.

[91] R. Shepard. Attention and the metric structure of the stimulus space. *Journal of Mathematical Psychology*, 1(1) :54–87, January 1964.

[92] Roger N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171 :701–703, 1971.

[93] Ken Shoemake. ARCBALL : a user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of the conference on Graphics interface '92*, pages 151–156, Vancouver, British Columbia, Canada, 1992. Morgan Kaufmann Publishers Inc.

[94] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual reality on a wim : Interactive worlds in miniature. pages 265–272, 1995.

[95] Paul S. Strauss and Rikk Carey. An object-oriented 3D graphics toolkit. *SIGGRAPH Comput. Graph.*, 26(2) :341–349, 1992.

[96] Robert J. Teather and Wolfgang Stuerzlinger. Guidelines for 3D positioning techniques. In *Proceedings of the 2007 conference on Future Play*, pages 61–68, Toronto, Canada, 2007. ACM.

[97] TUIO. TUIO. http ://www.tuio.org/, 2009.

[98] B. J. Unger, A. Nicolaidis, A. Thompson, R. L. Klatzky, R. L. Hollis, P. J. Berkelman, and S. Lederman. Virtual peg-in-hole performance using a 6-dof magnetic levitation haptic device : Comparison with real forces and with visual guidance alone. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, page 263, 2002.

[99] Arjen van Rhijn and Jurriaan D. Mulder. Spatial input device structure and bimanual object manipulation in virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 51–60, Limassol, Cyprus, 2006. ACM.

[100] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. Influence of degrees of freedom's manipulation on performances during orientation tasks in virtual reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*, pages 51–58, Kyoto, Japan, 2009. ACM.

[101] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. Dynamic decomposition and integration of degrees of freedom for 3-d positioning. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, VRST '10, pages 131–134, New York, NY, USA, 2010. ACM.

[102] Manuel Veit, Antonio Capobianco, and Dominique Bechmann. An experimental analysis of the impact of touch screen interaction techniques for 3-d positioning tasks. In *IEEE Conference On Virtual Reality*, VR '11, New York, NY, USA, 2011. ACM.

[103] Dan Venolia. Facile 3D direct manipulation. In *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 31–36, Amsterdam, The Netherlands, 1993. ACM.

[104] Daniel Vogel and Patrick Baudisch. Shift : a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 657–666, San Jose, California, USA, 2007. ACM.

[105] Daniel Vogel and Géry Casiez. Conté : multimodal input inspired by an artist's crayon. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 357–366, New York, NY, USA, 2011. ACM.

[106] Daniel Vogel, Matthew Cudmore, Géry Casiez, Ravin Balakrishnan, and Liam Keliher. Hand occlusion with tablet-sized direct pen input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 557–566, New York, NY, USA, 2009. ACM.

[107] Yanqing Wang, Christine L. MacKenzie, Valerie A. Summers, and Kellogg S. Booth. The structure of object transportation and orientation in human-computer interaction. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 312–319, Los Angeles, California, United States, 1998. ACM Press/Addison-Wesley Publishing Co.

[108] C. Ware. Using hand position for virtual object placement. *Vis. Comput.*, 6 :245–253, November 1990.

[109] Wayne Westerman. *Hand Tracking, Finger Identification and Chordic Manipulation on a Multi-Touch Surface*. PhD thesis, Faculty of the University of Delaware, 1999.

[110] Andrew D. Wilson. TouchLight : an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*, page 69–76, State College, PA, USA, 2004. ACM.

[111] Andrew D. Wilson. PlayAnywhere : a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, page 83–92, Seattle, WA, USA, 2005. ACM.

[112] Andrew D. Wilson. Simulating grasping behavior on an imaging interactive surface. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 125–132, New York, NY, USA, 2009. ACM.

[113] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 67–76, Monterey, CA, USA, 2008. ACM.

[114] Jeffrey A. Wolfeld, Jeffrey A. Wolfeld, Aravind K. Joshi, and Jeffrey A. Wolfeld. Real time control of a robot tactile sensor, 1981.

[115] Robert C. Zeleznik, Andrew S. Forsberg, and Paul S. Strauss. Two pointer input for 3D interaction. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 115–ff., Providence, Rhode Island, United States, 1997. ACM.

[116] Zhai. *Ph.D. Thesis : Human Performance in Six Degree of Freedom Input Control*. PhD thesis, University of Toronto, 1995.

[117] Shumin Zhai and Paul Milgram. Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, page 320–327, Los Angeles, California, United States, 1998. ACM Press/Addison-Wesley Publishing Co.