



L'interaction 3D adaptative : une approche basée sur les méthodes de traitement de données multi-capteurs

Pierre Boudoin

► To cite this version:

Pierre Boudoin. L'interaction 3D adaptative : une approche basée sur les méthodes de traitement de données multi-capteurs. Informatique [cs]. Université d'Evry-Val d'Essonne, 2010. Français. NNT : . tel-00553369v1

HAL Id: tel-00553369

<https://theses.hal.science/tel-00553369v1>

Submitted on 7 Jan 2011 (v1), last revised 7 Jan 2011 (v2)

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 D'ÉVRY-VAL D'ESSONNE

Laboratoire d'Informatique, Biologie Intégrative
et Systèmes Complexes

NNT : 2010 EVRY 25

Mémoire pour l'obtention du titre de Docteur
de l'Université d'Évry-Val d'Essonne
Spécialité : Sciences de l'ingénieur

L'INTERACTION 3D ADAPTATIVE : UNE APPROCHE BASÉE SUR LES MÉTHODES DE TRAITEMENT DE DONNÉES MULTI-CAPTEURS

Pierre Boudoin

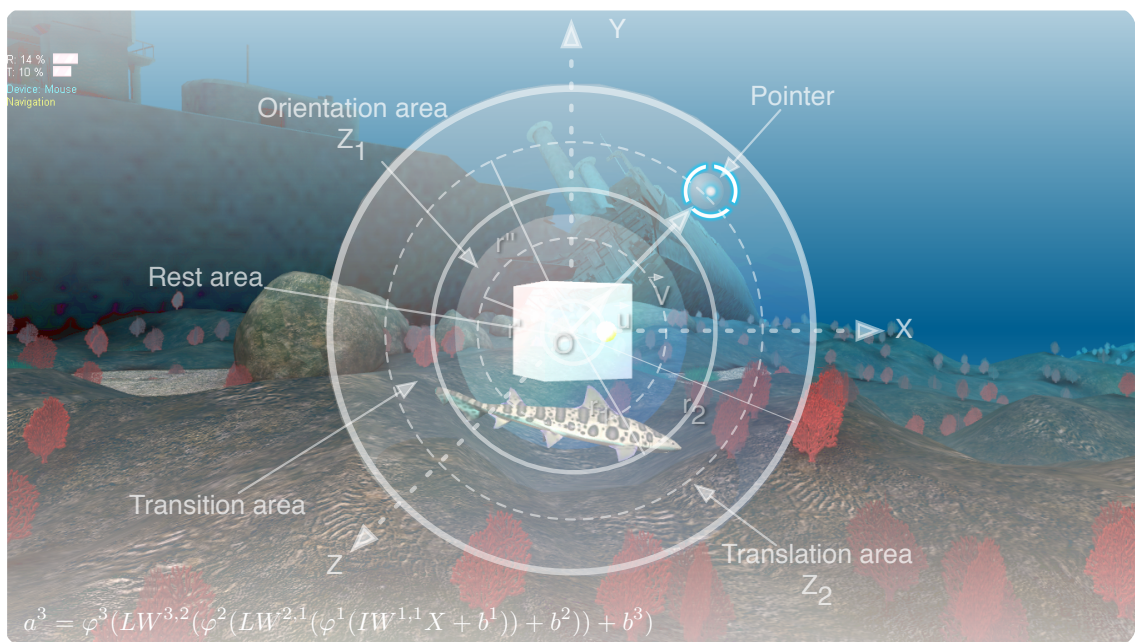
Soutenue le Mercredi 6 Octobre 2010

Jury

K. Madani	Prof.	Université de Paris 12	Rapporteur
G. Moreau	MdC.	Ecole Centrale de Nantes	Rapporteur
Y. Amirat	Prof.	Université de Paris 12	Examineur
H. Maaref	Prof.	Université d'Évry Val d'Essonne	Examineur
M. Mallem	Prof.	Université d'Évry Val d'Essonne	Examineur
S. Otmane	MdC	Université d'ÉvryVal d'Essonne	Examineur

L'INTERACTION 3D ADAPTATIVE : UNE APPROCHE BASÉE SUR LES MÉTHODES DE TRAITEMENT DE DONNÉES MULTI-CAPTEURS

PIERRE BOUDOIN



La simplicité est souvent synonyme d'efficacité.

— Leonard de Vinci

RÉSUMÉ

La réalité virtuelle est un domaine touchant à plusieurs disciplines. Par le biais de l'interaction 3D l'Homme peut accomplir des tâches dans un environnement virtuel en utilisant des techniques d'interaction 3D. Ces techniques sont souvent mono-tâches et s'appuient sur l'utilisation de matériel de réalité virtuelle bien spécifique. Le passage d'une tâche de l'interaction 3D à une autre est le plus souvent à la charge de l'utilisateur, ou bien du programmeur. Cependant de nombreux problèmes sont présents dans ces systèmes, dits de réalité virtuelle. En effet, des problèmes matériels le plus souvent dû aux technologies utilisées sont présents et peuvent induire un comportement erratique du système. De plus, il peut arriver que les techniques d'interaction 3D ne soient pas adaptées à l'application de réalité virtuelle, ou que celles-ci soient trop complexes à utiliser pour le novice. Tous ces problèmes nuisent à l'immersion de l'être humain dans l'environnement virtuel ainsi qu'aux performances de l'interaction 3D et donc à l'accomplissement de la tâche dans l'application de réalité virtuelle.

L'objectif de ce travail est de proposer un système d'interaction 3D adaptative. Par interaction 3D adaptative, on cherche à définir une interaction 3D qui soit continue tant au niveau des données qu'au basculement d'une tâche à l'autre. Nous avons donc modélisé et conçu un ensemble de composants pour accomplir cet objectif. Nous avons modélisé une technique d'interaction 3D pouvant être utilisée de manière continue même lors du basculement d'une tâche. Nous avons également conçu un système qui permet d'automatiser le basculement d'une tâche de l'interaction 3D vers une autre en estimant la tâche que souhaite accomplir l'utilisateur. Enfin, un dernier composant a pour rôle d'améliorer la précision et de garantir la continuité du tracking.

MOTS-CLÉS :

Réalité virtuelle, Interaction 3D, Calibration, Tracking, Réseau de neurones, Support Vector Regression.

ABSTRACT

Virtual reality is a field involving multiple disciplines. Through the 3D interaction of we can perform tasks in a virtual environment using 3D interaction techniques. These techniques are often mono-task and are based on the use of specific virtual reality devices. The transition from a 3D interaction task to another is usually to the responsibility of the user, or programmer. Indeed, due to the technologies used, hardware problems are present and can induce erratic behaviors from the system. Moreover, it may happen that 3D interaction techniques are not suited to the application of virtual reality, or that they are too complex to use for the novice. All these problems affect the immersion of

the user in the virtual environment and the performance of 3D interaction and thus, to accomplish the task in virtual reality application.

The aim of this work is to propose an adaptive 3D interaction system. Adaptive 3D interaction, defines a 3D interaction that is continuous in terms of data and switching from one task to another. So we have modeled and designed a set of components to accomplish this goal. We designed a 3D interaction technique which, can be used continuously even when switching to a task. We have also designed a system that automates the switch from a 3D interaction task to another task by estimating the task that the user wants to accomplish. The last component's role is to improve accuracy and ensure the continuity of tracking data.

KEYWORDS :

Virtual reality, 3D interaction, Calibration, Tracking, Neural networks, Support Vector Regression

REMERCIEMENTS

Je ne saurais commencer ce travail sans remercier toutes les personnes qui ont contribué à son aboutissement.

Je remercie donc, en tout premier lieu, M. Hichem Maaref, pour avoir accepté d'être le directeur de ma thèse, pour la confiance et la liberté de recherche qu'il m'a accordée ainsi que pour l'attention qu'il portait à mon travail.

Je remercie également M. Malik Mallem et M. Samir Otmane pour leur écoute et leurs conseils ainsi que pour m'avoir aidé dans toutes les démarches scientifiques.

Je remercie tous les doctorants, pour les moments partagés et les débats passionnés du midi, ainsi que tout le personnel administratif du laboratoire.

Je remercie bien évidemment tous mes collègues de la salle Ax15, qui sont maintenant devenus des amis et avec lesquels j'ai partagé tant de bons moments : merci Mouna, Nader, Imane, Mahmoud et Christophe ainsi que Jean-Yves et Christophe M.

Ma famille et mes amis ont bien évidemment droit, eux aussi, à mes remerciements pour tout leur soutien, tant dans les moments agréables que désagréables.

Et un grand merci à Charlène, qui dans mes moments de doutes, a su, avec sa grande sagesse, trouver les mots justes et m'a encouragé tout au long de cette dernière année de thèse. Je lui dois ma persévérance.

Merci à tous.

TABLE DES MATIÈRES

INTRODUCTION GÉNÉRALE	1
I ETAT DE L'ART	5
1 L'INTERACTION 3D EN RÉALITÉ VIRTUELLE	7
1.1 Introduction	7
1.2 La réalité virtuelle	8
1.3 L'interaction 3D	11
1.3.1 Présentation	11
1.3.2 Trèfle fonctionnel de l'interaction 3D	11
1.3.3 Les tâches de l'interaction 3D en réalité virtuelle	12
1.3.4 Lien entre les tâches de l'interaction 3D	14
1.3.5 Les techniques d'interaction 3D courantes en réalité virtuelle	14
1.4 Les assistances à l'utilisateur pour l'interaction 3D	26
1.4.1 L'ontologie	26
1.4.2 La conscience de l'interaction	27
1.4.3 Les guides virtuels	28
1.4.4 Les guides haptiques	29
1.5 Les architectures multi-capteurs et multimodales	30
1.5.1 Présentation	30
1.5.2 Architectures génériques	32
1.5.3 Suivi de la position multi-capteurs	32
1.5.4 Ontologie spatiale et architecture d'interaction 3D multimodale	34
1.5.5 Désambiguïsation mutuelle pour l'interaction 3D multimodale	38
1.5.6 VR Juggler	39
1.5.7 VRPN:Virtual Reality Peripheral Network	40
1.6 Conclusion	40
2 LES PÉRIPHÉRIQUES ET CAPTEURS D'INTERACTION UTILISÉS EN RÉALITÉ VIRTUELLE	41
2.1 Introduction	41
2.2 Les périphériques de tracking	41
2.2.1 Les systèmes de tracking optique	42
2.2.2 Le tracking magnétique	44
2.2.3 Les centrales inertielles et les accéléromètres	45
2.2.4 Le tracking 3D à ultrasons	46
2.3 Les gants de données	46
2.4 Les périphériques à retour d'effort	48
2.4.1 Les bras haptiques	48
2.4.2 Le SPIDAR	49
2.5 Les périphériques hybrides	49
2.6 Conclusion	50
3 LA CALIBRATION DES SYSTÈMES DE TRACKING 3D EN RÉALITÉ VIRTUELLE	53
3.1 Introduction	53
3.2 La calibration d'un système de tracking	53
3.3 Les protocoles expérimentaux de calibration des systèmes de tracking	54

3.4	Les méthodes de régression pour la calibration	55
3.4.1	Les méthodes de régression linéaire	55
3.4.2	Les méthodes de régression non-linéaire	56
3.5	La caractérisation d'un système de tracking	65
3.5.1	Evaluation des distorsions d'un système de tracking	65
3.5.2	Quelques moyens de représentation des distorsions d'un système de tracking	65
3.6	Quelques exemples de calibration de systèmes de tracking	68
3.6.1	La correction de distorsions statiques sur les systèmes de tracking 3D	68
3.6.2	Système de tracking hybride	70
3.7	Conclusion	71
II	CONTRIBUTIONS	73
4	PROPOSITION D'UN SYSTÈME D'INTERACTION 3D ADAPTATIVE	75
4.1	Introduction	75
4.2	Démarche scientifique:	75
4.3	Architecture proposée	76
4.4	La technique d'interaction 3D : Fly-Over	78
4.4.1	Introduction	78
4.4.2	Spécification du modèle générique de Fly-Over	78
4.4.3	Fly-Over pour la tâche de navigation : Fly-Over-N	80
4.4.4	Fly-Over pour la tâche de sélection : Fly-Over-S	84
4.4.5	Fly-Over pour la tâche de manipulation : Fly-Over-M	86
4.4.6	Fly-Over pour la tâche de contrôle d'application : Fly-Over-CA	86
4.4.7	Discussion autour des tâches de Fly-Over	88
4.4.8	Les assistances multimodales de Fly-Over pour l'utilisateur	88
4.5	L'unité d'interaction 3D adaptative	95
4.5.1	Introduction	95
4.5.2	Paramètres observables	95
4.5.3	Système de décision	96
4.6	L'unité de correction et combinaison des données: UCCD	97
4.6.1	Contexte dans lequel se positionne l'UCCD	98
4.6.2	L'UCCD dans un système de réalité virtuelle	99
4.6.3	Les différents composants de l'UCCD	99
4.6.4	Algorithme de traitement	103
4.6.5	Relations entre l'UCCD, l'UI3D-A et Fly-Over	104
4.7	Bilan	105
4.7.1	Notre système dans différentes applications de réalité virtuelle	105
4.7.2	Conclusion	106
5	ETUDE ET CALIBRATION DU SPIDAR	109
5.1	Introduction	109
5.2	Caractérisation du SPIDAR à l'aide du tracking optique	109
5.2.1	Contexte	109
5.2.2	Problèmes de conception rencontrés dans le SPIDAR	110
5.2.3	Hypothèses de travail	111
5.2.4	Protocole expérimental mis en place	112
5.2.5	Incertitudes sur la position du SPIDAR	113
5.3	Initialisation multimodale du SPIDAR	114
5.3.1	Contexte	114

5.3.2	Solution proposée pour minimiser l'influence des problèmes de conception	115
5.4	Calibration du SPIDAR par une régression linéaire	117
5.4.1	Présentation	117
5.4.2	Résultats obtenus par une régression linéaire (interpolation polynomiale d'ordre 1)	118
5.5	Calibration du SPIDAR à l'aide d'un réseau de neurones	120
5.5.1	Présentation	120
5.5.2	Configuration du réseau de neurones	121
5.5.3	Apprentissage du réseau de neurones	121
5.5.4	Nombre optimal de neurones dans la couche cachée	122
5.5.5	Performances du réseau de neurones en apprentissage	124
5.5.6	Résultats du réseau de neurones en généralisation	127
5.5.7	Conclusion sur l'utilisation d'un réseau de neurones	129
5.6	Calibration du SPIDAR par les SVRs	129
5.6.1	Présentation	129
5.6.2	Choix des paramètres pour les SVRs	129
5.6.3	Performances des SVRs sur l'apprentissage	129
5.7	Méthode de calibration retenue et application pratique de la calibration	131
5.7.1	Tableau comparatif des différentes méthodes de correction	132
5.7.2	Application pratique	132
5.7.3	Logigramme de basculement automatique:	133
5.7.4	Résultats:	134
5.8	Conclusion	135
6	EVALUATION DE L'UNITÉ D'INTERACTION 3D ADAPTATIVE	137
6.1	Scénario d'évaluation mis en place	137
6.2	Profils des participants à l'évaluation	138
6.3	Evaluation objective	139
6.3.1	Déroulement de l'évaluation objective	139
6.3.2	Paramètres observés pour l'évaluation	141
6.3.3	Résultats de l'évaluation objective et interprétation	141
6.3.4	Soumission du questionnaire d'évaluation	146
6.3.5	Etude de l'utilisabilité	146
6.3.6	Etude du confort	148
6.4	Observations constatées pendant les évaluations	149
6.5	Conclusion	150
III	CONCLUSION & PERSPECTIVES	151
IV	BIBLIOGRAPHIE	159
V	ANNEXES	169
A	QUESTIONNAIRE	171

TABLE DES FIGURES

Figure 1	Une scène issue du film <i>Minority Report</i>	7
Figure 2	Le continuum réel-virtuel adapté de [Milgram and Kishino, 1994].	8
Figure 3	Les trois médiations du modèle en réalité virtuelle proposé par [Tisseau, 2001] Tisseau	9
Figure 4	Les trois I de Burdea and Coiffet [Burdea and Coiffet, 1993].	9
Figure 5	La présence et l'autonomie en réalité virtuelle, proposé par Tisseau [Tisseau, 2004].	10
Figure 6	Schéma de Fuchs and Moreau représentant les 3 niveaux d'interaction en réalité virtuelle [Fuchs and Moreau, 2003].	11
Figure 7	Trèfle fonctionnel de l'interaction 3D extrait de [Otmane et al, 2007].	12
Figure 8	Lien entre les tâches de l'interaction 3D définis par Bowman [Bowman, 1999].	14
Figure 9	La technique direction du regard [Mine, 1995].	16
Figure 10	Fonctionnement du mode orbital [Mine, 1995].	16
Figure 11	Une personne entrain d'utiliser la technique : <i>Déplacement sur une carte</i> (photo issue de [Bowman, 1999]).	17
Figure 12	Image illustrant la technique <i>Magallanes</i> avec l'interface graphique attachée à la scène [Abásolo and Della, 2007].	17
Figure 13	Contrôle de la vitesse par la main [Mine, 1995].	18
Figure 14	Le <i>Navidget</i> (à gauche) et le <i>Navidget</i> en utilisation dans une application de réalité virtuelle (à droite) [Hachet et al, 2009].	19
Figure 15	Comparaison des approches des techniques de sélection locale et à distance [Mine, 1995].	19
Figure 16	La technique d'interaction 3D de la <i>Main virtuelle simple</i>	20
Figure 17	La technique d'interaction 3D du <i>Raycasting</i>	20
Figure 18	La technique d'interaction 3D <i>Occlusion</i>	20
Figure 19	Représentation de la technique <i>Go-Go</i> et sa fonction mathématique définissant l'extension de la main virtuelle [Poupyrev et al, 1996].	21
Figure 20	Classification par décomposition des tâches des techniques de sélection extrait de [Bowman, 1999].	21
Figure 21	Chronologie de l'utilisation de la technique HOMER [Bowman, 1999].	22
Figure 22	Aperçu de la technique <i>Monde en miniature</i> [Pausch et al, 1995].	23
Figure 23	Zones d'interaction de la technique <i>Follow-me</i>	24
Figure 24	Un exemple de menu 2D transformés afin de les rendre compatibles dans un espace 3D [Bowman, 1999].	24
Figure 25	Le <i>ViewCube</i> : en cliquant sur la face avant du cube, le widget (cube) ainsi que la scène 3D s'orientent de manière à ce disposer en vue de face [Khan et al, 2008].	25
Figure 26	Exemple d'ontologie pour le placement spatial d'un objet [Irawati et al, 2005].	27
Figure 27	Un exemple de conscience de l'interaction, illustrant le focus de l'utilisateur, c'est-à-dire le volume dans lequel seront situés les objets qui seront susceptibles de l'intéresser [Otmane et al, 2007].	28

Figure 28	Application d'un guide virtuel pour guider l'utilisateur à saisir un objet avec une pince dans le cadre de la téléopération. Le guide virtuel apparaît en vert sur la photo [Otmane and Davesne, 2009].	28
Figure 29	Représentation des guides répulsifs placés sur les différentes faces internes d'un parallélépipède [Otmane and Davesne, 2009].	29
Figure 30	Application des guides répulsifs à une application de téléopération. Le guide répulsif apparaît comme un cube blanc translucide [Otmane and Davesne, 2009].	30
Figure 31	Schéma représentant le guide haptique proposé par Ullah et al [Ullah et al, 2008].	30
Figure 32	Problématique de l'interaction 3D multi-capteurs.	31
Figure 33	Le <i>Virtual Reality System</i> et ses extensions [Steinicke et al, 2005].	32
Figure 34	La pile d'acquisition conçue par Leonhardt and Magee [Leonhardt and Magee, 1998].	33
Figure 35	Un exemple de structure récursive [Leonhardt and Magee, 1998].	33
Figure 36	Architecture d'interaction 3D multimodale adaptée [Irawati et al, 2006].	34
Figure 37	Relations périphériques - serveur de périphériques - gestionnaire de périphériques - gestionnaire d'interaction, adapté de [Irawati et al, 2006].	37
Figure 38	La boucle d'interaction adaptée de [Irawati et al, 2006].	37
Figure 39	Architecture générale [Kaiser et al, 2003].	38
Figure 40	Architecture de VR Juggler.	39
Figure 41	Le fonctionnement du système optique ART.	43
Figure 42	Le LASER utilisé avec une lentille diffractante [Chávez et al, 2008].	44
Figure 43	Schéma d'un pointeur LASER se réfléchissant sur un miroir [Matveyev and Matveyev, 2004].	44
Figure 44	Un exemple de dispositif de tracking magnétique, le <i>Fastrack</i> de Polhemus.	45
Figure 45	Un exemple d'accéléromètre.	45
Figure 46	Le Data Glove Ultra de Fifth Dimension Technologies.	46
Figure 47	Le gant CyberGlove de Cyber Gloves Systems.	46
Figure 48	Le gant CyberGrasp de Cyber Gloves Systems.	47
Figure 49	Le gant CyberTouch de Cyber Gloves Systems.	47
Figure 50	L'accelelglove [Hernandez-Rebollar and Kyriakopoulos, 2002].	47
Figure 51	Un bras haptique PHANTOM de Sensable Technologies.	48
Figure 52	Le Virtuouse de Haption.	48
Figure 53	Structure d'un SPIDAR.	49
Figure 54	La Wiimote de Nintendo® en main.	50
Figure 55	Les composants du BlueWand [Fuhrmann et al, 2003].	50
Figure 56	Disposition du plan de calibration de Leotta [Leotta, 2004].	54
Figure 57	L'objet étalon utilisé par Barratt et al [Barratt et al, 2001].	55
Figure 58	Un neurone simple sans biais.	57
Figure 59	Un neurone simple avec biais.	57
Figure 60	Un neurone avec une entrée vectorielle.	58
Figure 61	Un réseau de neurones à 3 couches.	58
Figure 62	Représentation des distorsions sous la forme de vecteurs 3D dans l'espace de travail adapté de [Ghazisaedy and Adamczyk, 1995].	66

Figure 63	Représentation de l'erreur en fonction de la distance des sources dans les travaux de Bryson [Bryson, 1992].	66
Figure 64	Représentation de l'erreur en fonction de la distance des sources dans les travaux de Ghazisaedy and Adamczyk [Ghazisaedy and Adamczyk, 1995].	67
Figure 65	Superposition des positions relevées avant et après calibration [Ikits et al, 2003].	67
Figure 66	Représentation des erreurs pour chacun des points de la grille de calibration [Ikits et al, 2003]	68
Figure 67	Le dispositif de tracking hybride mis en place par Barratt et al [Barratt et al, 2001].	68
Figure 68	Résultats obtenus par Kindratenko and Sherman via l'utilisation d'un réseau de neurones par rapport à d'autre méthodes [Kindratenko and Sherman, 2005].	69
Figure 69	Avant et après correction du tracking par la méthode proposée par Livingston [Livingston, 1997].	69
Figure 70	Logigramme du système de tracking hybride proposé par Birkfellner et al [Birkfellner et al, 1998].	70
Figure 71	Réflexion sur les différents composants nécessaires pour répondre à l'objectif.	76
Figure 72	Cycles de l'architecture.	77
Figure 73	Modèle simplifié de Fly-Over - N.	80
Figure 74	Evolution du modèle de Fly-Over - N avec l'ajout d'une zone de transition entre la zone de rotation et de translation.	82
Figure 75	Exemple de variation des fonctions poids C_R et C_T	83
Figure 76	Disposition du pointeur lors de la selection.	84
Figure 77	Exemple de sélection avec occultation - L'utilisateur peut sélectionner un objet qui est situé derrière un autre en jouant sur la profondeur du spot.	85
Figure 78	Aperçus des gestes de catégorie secousses.	86
Figure 79	Aperçus des gestes de catégorie balayage.	87
Figure 80	Aperçu de la boussole par rapport à la position du pointeur.	89
Figure 81	Les différents états du pointeur de Fly-Over en fonction du contexte.	90
Figure 82	Interface graphique de Fly-Over.	90
Figure 83	Interface graphique de Fly-Over lors de la tâche de sélection.	91
Figure 84	Interface graphique de Fly-Over lors de la tâche de manipulation.	91
Figure 85	Variation du gain dans l'espace : Plus la différence de rayon entre deux cercles est grande, plus le gain est grand.	93
Figure 86	Les 4 paramètres utilisés par l'unité d'interaction 3D adaptative.	95
Figure 87	Règles pour basculer entre les tâches de l'interaction 3D.	96
Figure 88	Architecture classique de système multi-capteurs en réalité virtuelle.	98
Figure 89	Architecture de système multi-capteurs en réalité virtuelle avec couche d'abstraction.	98
Figure 90	Architecture de système multi-capteurs en réalité virtuelle avec UCCD.	99
Figure 91	Description de l'initialisation des périphériques.	100
Figure 92	Description des périphériques virtuels.	100
Figure 93	Description des événements virtuels.	101
Figure 94	L'unité de correction et combinaison des données.	101

Figure 95	Algorithme de traitement simplifié.	103
Figure 96	Interaction entre le l'UCCD et Fly-Over.	104
Figure 97	Différentes applications de réalité virtuelle utilisant la technique d'interaction Fly-Over.	106
Figure 98	Les différentes couches de notre système d'interaction 3D adaptative permettant à une personne d'interagir en continu.	107
Figure 99	Vue détaillée d'un moteur du SPIDAR et du guide d'enroulement. . .	110
Figure 100	Représentation de l'espace de travail (en vert) du SPIDAR au sein de sa structure (en noir).	111
Figure 101	Disposition du matériel.	112
Figure 102	Cadre expérimental du protocole. A gauche, un utilisateur est en train de réaliser la prise des positions. A droite, une vue détaillée de ce que perçoit l'utilisateur (en rouge l'effecteur virtuel que contrôle l'utilisateur).	113
Figure 103	Représentation de l'erreur absolue dans le volume de travail du SPIDAR.	114
Figure 104	Zones d'initialisation et cadre expérimental.	115
Figure 105	Algorithme du processus d'initialisation du SPIDAR.	117
Figure 106	Erreurs absolues sur la position du SPIDAR par rapport à la distance à laquelle la mesure a été prise du point d'initialisation, avant et après calibration par interpolation polynomiale d'ordre 1.	118
Figure 107	Répartition de l'erreur absolue sur la position du SPIDAR dans l'espace de travail.	119
Figure 108	Erreur absolue sur la position du SPIDAR.	120
Figure 109	Réseau de neurones utilisé pour la correction du SPIDAR.	121
Figure 110	Erreur absolue moyenne sur la position par rapport au nombre de neurones sur la couche cachée.	122
Figure 111	Performance du réseau de neurones utilisé pour la calibration.	123
Figure 112	réseau de neurones utilisé pour la calibration du SPIDAR	123
Figure 113	Régression en fonction des bases de données utilisées.	124
Figure 114	Erreurs absolues sur la position du SPIDAR par rapport à la distance à laquelle la mesure a été prise du point d'initialisation, avant et après calibration par le réseau de neurones.	125
Figure 115	Représentation de l'erreur absolue dans l'espace sur la position, avec une plage d'erreur réduite de moitié.	126
Figure 116	Erreur absolue sur la position du SPIDAR avant et après calibration par le réseau de neurones.	126
Figure 117	Erreur absolue moyenne sur la position du SPIDAR avant et après le réseau de neurones sur la base de données 1.	127
Figure 118	Erreur absolue moyenne sur la position du SPIDAR avant et après le réseau de neurones sur la base de données 2.	128
Figure 119	Erreurs absolues sur la position du SPIDAR par rapport à la distance à laquelle la mesure a été prise du point d'initialisation, avant et après calibration par les SVRs.	130
Figure 120	Répartition de l'erreur absolue sur la position du SPIDAR dans l'espace de travail.	130
Figure 121	Erreur absolue sur la position du SPIDAR avant et après calibration par les SVRs.	131

Figure 122	Etude comparative des erreurs sur la calibration en fonction des différentes méthodes.	132
Figure 123	Algorithme de basculement entre les deux systèmes.	133
Figure 124	Exemple d'une trame du tracking hybride mis en place	134
Figure 125	Informations sur le profil des participants 1/2	138
Figure 126	Informations sur le profil des participants 2/2	139
Figure 127	Configuration des boutons du flystick pour la seconde phase de l'évaluation.	140
Figure 128	La phase de navigation directe de point en point avec Fly-Over.	140
Figure 129	Temps moyens de la réalisation de l'épreuve demandée en fonction des essais.	143
Figure 130	Temps moyen de la réalisation de l'épreuve - Groupe des experts.	144
Figure 131	Temps moyen de la réalisation de l'épreuve - Groupe des intermédiaires.	145
Figure 132	Temps moyen de la réalisation de l'épreuve - Groupe des novices.	146
Figure 133	Jugement de la fiabilité de l'unité d'interaction 3D adaptative par les participants.	147
Figure 134	Jugement des participants sur la difficulté à accomplir l'épreuve de l'évaluation.	147
Figure 135	Jugement des participants sur la praticité à basculer d'une tâche de l'interaction 3D à une autre.	148
Figure 136	Etat de fatigue des participants avant et après l'évaluation (recueillis à partir des questionnaires.)	148
Figure 137	Jugement des participants sur l'interaction 3D.	149

LISTE DES TABLEAUX

Table 1	Résumé des différents gains en fonction de la tâche de l'interaction 3D et de la position du pointeur.	93
Table 2	Table de décision pour passer d'une tâche de l'interaction 3D à une autre.	97
Table 3	Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par régression linéaire.	120
Table 4	Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par un réseau de neurones.	127
Table 5	Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par le réseau de neurones sur la base de données 1.	128
Table 6	Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par le réseau de neurones sur la base de données 2.	128
Table 7	Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par les SVRs.	131
Table 8	Comparaison des valeurs caractéristiques des erreurs absolues.	132

Table 9	Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Tous les participants.	141
Table 10	Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Groupe des experts.	142
Table 11	Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Groupe des intermédiaires.	142
Table 12	Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Groupe des novices.	143
Table 13	Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Tous les participants.	143
Table 14	Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Groupe des experts.	144
Table 15	Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Groupe des intermédiaires.	145
Table 16	Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Groupe des novices.	146

INTRODUCTION GÉNÉRALE

CONTEXTE

L'interaction 3D en réalité virtuelle permet au travers de métaphores et paradigmes d'interagir avec un environnement créé numériquement soit à partir de données réelles, soit complètement conçu par ordinateur. Les applications de l'interaction 3D en réalité virtuelle sont multiples : consultation d'informations en 3D, représentation d'entités réelles ne pouvant être observées ou représentées en condition réelles, réalisation de simulation et création d'œuvres numériques.

Cette thèse propose une approche combinant les recherches en **réalité virtuelle** et en **traitement de données** pour répondre à la problématique de l'**interaction 3D adaptative et continue**. La Réalité Virtuelle permet la prise en compte des tâches et des interfaces d'interactions 3D (ensemble de capteurs). Le traitement de données offre des approches permettant de gérer les données issues des systèmes multi-capteurs.

PROBLÉMATIQUE

Dans les systèmes d'interaction 3D, l'utilisateur est dans un environnement composé de multiples capteurs qui peuvent être également embarqués sur lui, dont lequel tous ces faits et gestes sont mesurés et détectés. Un tel système s'apparente à un système multi-capteurs dont les objectifs sont multiples :

- Pallier les limitations technologiques ou physiques : Chaque capteur repose sur une technologie qui permet de mesurer certaines caractéristiques d'un phénomène physique ce qui peut induire des erreurs ou une mauvaise interprétation des données provenant du capteur.
- Créer de l'information : chaque capteur permet de récupérer un certain type de données en cumulant plusieurs capteurs et ainsi plusieurs données. On peut donc créer une nouvelle information à l'aide de méthodes de fusion de données.
- Aider à la décision : les méthodes de fusion de données peuvent aussi être utilisées dans l'aide à la décision. Dans ce cas la décision prise doit être la meilleure possible compte tenu des informations recueillies.

Si du point de vue du rendu 3D de ces environnements virtuels, les limites sont sans cesse repoussées et de moins en moins contraignantes, dues à la montée en puissance des ordinateurs, il reste encore, beaucoup de travail à accomplir pour l'interaction avec ceux-ci. Il existe un grand nombre de techniques d'interaction 3D (logiciel) et de périphériques (matériel), mais il est toujours difficile de proposer un système qui fonctionne avec le plus grand nombre d'applications de réalité virtuelle, et qui soit accessible en termes d'ergonomie, d'efficacité et de simplicité à l'utilisation.

Pour répondre à cette problématique, ce travail propose d'aborder tous les aspects de l'interaction 3D, c'est-à-dire de la gestion des périphériques et de leurs données,

jusqu'aux techniques de l'interaction 3D. Tous ces aspects sont regroupés au sein d'une seule architecture, conçue de manière à ce que les composants agissent de concert. L'ensemble formant un système d'interaction 3D adaptative et ergonomique.

L'objectif de ce travail est de proposer un **système d'interaction 3D adaptative**. Par interaction 3D adaptative, on cherche à définir une interaction qui soit **continue** (qu'elle soit matérielle ou logicielle) et qui s'**adapte** au contexte de l'environnement réel et virtuel.

ORGANISATION DE LA THÈSE

Notre travail de recherche, réalisé durant cette thèse, se place dans le cadre de l'interaction 3D en général. En effet, il couvre l'ensemble des aspects d'un système d'interaction 3D. On distingue deux parties dans ce manuscrit :

- La première regroupe l'état de l'art des recherches effectuées sur les concepts, méthodes et outils utilisés dans les différents domaines que rassemble un système d'interaction 3D c'est-à-dire : La réalité virtuelle et l'interaction 3D, les périphériques de tracking en réalité virtuelle ainsi que le domaine du traitement de données pour la calibration de périphériques de tracking.
- La seconde partie, présente le travail personnel effectué tout au long de cette thèse, qui se consacre à la conception d'un système entier permettant à un utilisateur d'interagir en 3D de manière continue. L'ensemble de ce travail, se décline en trois contributions : La proposition d'une technique d'interaction 3D innovante, la conception d'un module de traitement de données et d'un module d'interaction 3D adaptative permettant d'automatiser certains aspects de l'interaction 3D.

Chapitre 1 : L'interaction 3D en réalité virtuelle

Le premier chapitre est consacré à la présentation du domaine de la réalité virtuelle et de l'interaction 3D ainsi qu'aux concepts qui leurs sont rattachés : comme les techniques, métaphores et paradigmes couramment utilisées dans ce domaine. Dans ce chapitre, sont également présentés quelques architectures de systèmes multimodaux afin d'étudier la manière dont les différents composants d'un système d'interaction 3D en réalité virtuelle s'agencent.

Chapitre 2 : Les périphériques et capteurs d'interaction utilisés en réalité virtuelle

Dans le second chapitre, nous nous intéressons aux périphériques et capteurs utilisés en réalité virtuelle afin de réaliser le suivi d'un objet, d'un corps ou d'une partie du corps humain. Nous étudions dans quelles mesures ces périphériques sont la clé de voute de tout système de réalité virtuelle. Nous étudions également, les technologies utilisées, leurs avantages et inconvénients ainsi que les problématiques qui en découlent.

Chapitre 3 : La calibration des systèmes de tracking 3D en réalité virtuelle

Le troisième chapitre expose le problème de la calibration des systèmes de tracking en 3 dimensions et la présentation de méthodes de régression et de protocoles expérimentaux utilisés pour palier les problèmes constatés dans le deuxième chapitre. Nous abordons les difficultés relevant des conceptions des périphériques, des technologies utilisées et des protocoles expérimentaux que l'on peut trouver dans la littérature pour les étudier ainsi que les différentes approches pour les corriger.

Chapitre 4 : Proposition d'un système d'interaction 3D adaptative

Le quatrième chapitre présente notre apport sous la forme d'une architecture de système d'interaction 3D adaptative basé sur 3 composantes:

- une **technique d'interaction 3D** innovante permettant de réaliser en continu et de manière intuitive les différentes tâches de l'interaction 3D (navigation, sélection et manipulation) avec n'importe quel périphérique de pointage 3D [6, 7];
- une **unité d'interaction 3D adaptative** qui fait office de contrôle d'application en estimant la tâche de l'interaction 3D que souhaite réaliser l'utilisateur et en basculant automatiquement vers celle-ci;
- une **unité de traitement de données** qui est un outil permettant de traiter à la volée les données issues des différents périphériques du système [3, 4, 5, 8].

Chapitre 5 : Étude et calibration du SPIDAR

Le cinquième chapitre traite de la calibration d'un système de tracking toujours disponible mais peu précis par rapport à un système de tracking précis mais qui est sujet à occultation. Dans notre cas, ceci consiste à calibrer un SPIDAR¹ (un périphérique électromécanique, permettant le suivi d'une main avec un retour d'effort) par rapport à un système de tracking optique, afin d'améliorer sa précision en position et ainsi assurer le bon fonctionnement de la technique d'interaction 3D présenté dans le chapitre 4. En effet, celle-ci nécessite un vecteur position qui soit continu dans le temps et dans l'espace. Dans un premier temps, une étude de la conception du SPIDAR est réalisée afin de le caractériser. Puis, une modélisation, basée sur deux méthodes de régression non-linéaires (réseau de neurones [2] et support vector regression [1]), est proposée pour calibrer ce périphérique par rapport au tracking optique.

Chapitre 6 : Évaluation de l'unité d'interaction 3D adaptative

Enfin le sixième chapitre, est consacré aux évaluations de l'unité d'interaction 3D adaptative ainsi qu'à l'analyse des résultats afin de démontrer les bénéfices apportés par celle-ci dans la boucle d'interaction 3D. Ces résultats démontrent que l'unité d'interaction 3D adaptative est presque tout autant efficace que la décision humaine pour ce qui est de la gestion du basculement d'une tâche de l'interaction 3D vers une autre.

1. SPace Interface Device for Artificial Reality

Enfin, nous terminons ce mémoire par une conclusion comportant un bilan des différents aspects traités ainsi que des perspectives.

PUBLICATIONS SCIENTIFIQUES

- [1] Boudoin P, Maaref H, Otmane S, Mallem M. *SPIDAR Calibration using Support Vector Regression*. IPTA'10 : Proceedings of the International Conference on Image Processing Theory, Tools and Applications (2010).
- [2] Boudoin P, Maaref H, Otmane S, Mallem M. *SPIDAR Calibration based on Neural Networks versus Optical Tracking*. ANNIIP'10 : Proceedings of the 6th International Workshop on Artificial Neural Networks and Intelligent Information Processing (2010).
- [3] Boudoin P, Otmane S, Mallem M, Maaref H. *A Framework for Designing Adaptative Systems in VR Applications*. CCNC'09 : Electronic Proceedings of the 6th IEEE Consumer Communications and Networking Conference (2009).
- [4] Boudoin P, Otmane S, Mallem M, Maaref H. *A-3DI core : a framework for adding adaptative behaviour into VR applications*. VRCAI'08 : Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (2008).
- [5] Boudoin P, Otmane S, Mallem M, Maaref H. *An easy-to-use Framework to integrate Data Processing and Data Fusion in VR Applications*. ICAT'08 : Proceedings of the 18th International Conference on Artificial Reality and Telexistence (2008). pp. 129-136.
- [6] Boudoin P, Otmane S, Mallem M. *Design of a 3D Navigation Technique Supporting VR Interaction*. CISA'08 : Proceedings of the 1st Mediterranean Conference on Intelligent Systems and Automation (2008). Vol 1019. pp. 149-153.
- [7] Boudoin P, Otmane S, Mallem M. *Fly Over, a 3D Interaction Technique for Navigation in Virtual Environments Independent from Tracking Devices*. VRIC'08 : Proceeding of the Virtual Reality International Conference (2008) pp. 7-13.
- [8] Boudoin P, Domingues C, Otmane S, Ouramdane N, Mallem M. *Towards multimodal human-robot interaction in large scale virtual environment*. HRI'08 : Proceedings of the 3rd international conference on Human robot interaction (2008) pp. 359-366.

Partie I

ETAT DE L'ART

L'INTERACTION 3D EN RÉALITÉ VIRTUELLE

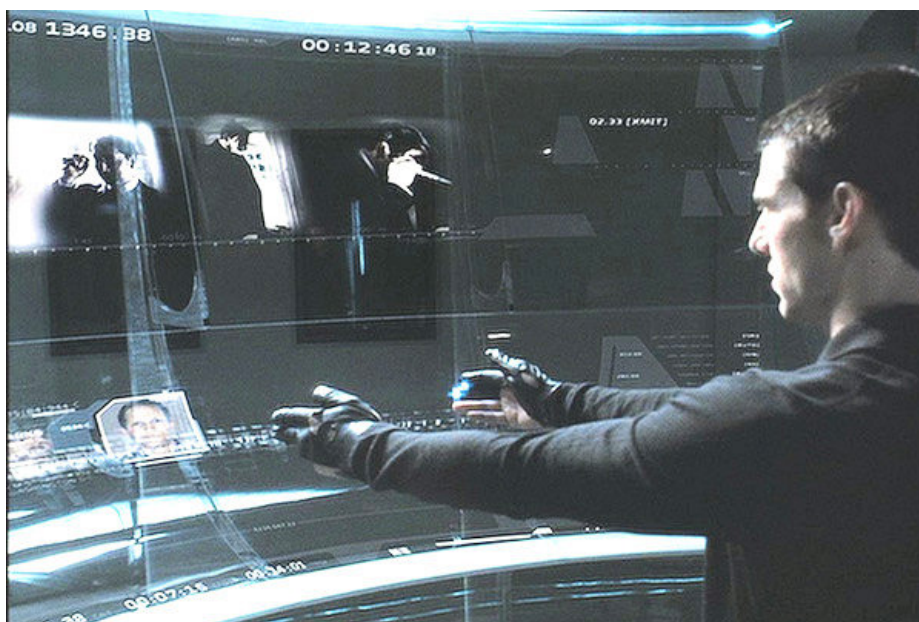


Figure 1: Une scène issue du film *Minority Report*.

1.1 INTRODUCTION

Depuis la nuit des temps, l'Homme a toujours utilisé de son imagination pour s'évader, se divertir ou tout simplement pour expliquer des choses qui ne lui étaient pas visibles. Au fur et à mesure des découvertes scientifiques et de l'évolution des technologies, l'Homme a commencé à utiliser des supports pour aider son imagination ou même l'initier. Cela a commencé avec la peinture, la musique, les livres, puis sont venus la photographie et le cinéma. Cependant tous ces supports sont figés, l'Homme ne peut pas interagir avec eux, il est seulement spectateur. C'est une des raisons pour laquelle au cours du 19^{ème} siècle l'Homme a commencé à développer des technologies permettant de s'immerger dans ces expériences audiovisuelles. C'est ainsi qu'est apparu le domaine de la réalité virtuelle.

La réalité virtuelle est un domaine que l'on qualifie de jeune, mais il est cependant basé sur des technologies et des concepts qui sont anciens. Ce domaine permet à l'être humain de vivre des expériences basées sur une représentation partielle et artificielle de la réalité. Dans cette représentation, il sera à même de réaliser des tâches ou bien de percevoir voire ressentir des choses qu'il ne lui aurait pas été permis dans la réalité.

Ainsi la réalité virtuelle peut être utilisée dans de nombreux domaines. Elle peut permettre à l'être humain d'interagir avec des éléments avec lesquels dans la réalité il lui est impossible de le faire (amarrage de molécule). Elle peut également servir à former l'être humain à réaliser des tâches qui dans la réalité seraient trop coûteuses à réaliser, ou bien, qui nécessiteraient de trop gros moyens (simulateur de vol, simulateur de scénario de catastrophe). Mais, elle peut tout simplement servir à la distraction (effets spéciaux dans le cinéma) ou bien à l'art. Elle peut également permettre de sécuriser certains gestes en simulant d'abord l'interaction sur un modèle virtuel et seulement après vérification, qu'aucun risque n'est encouru pour l'appliquer ensuite au modèle réel (domaine de la télé-robotique).

Milgram and Kishino ont défini un continuum allant du réel au virtuel (figure 2) avec la part de chacun des deux mondes [Milgram and Kishino, 1994]. Ils définissent ainsi la réalité mixte (RM) qui est située à la limite de ces deux mondes. On y retrouve le domaine de la réalité augmentée (RA), qui consiste en la superposition d'informations supplémentaires sur des éléments réels, le plus souvent au travers d'un flux vidéo. La virtualité augmentée, prend la problématique dans l'autre sens, c'est-à-dire en superposant des éléments réels sur un support virtuel.

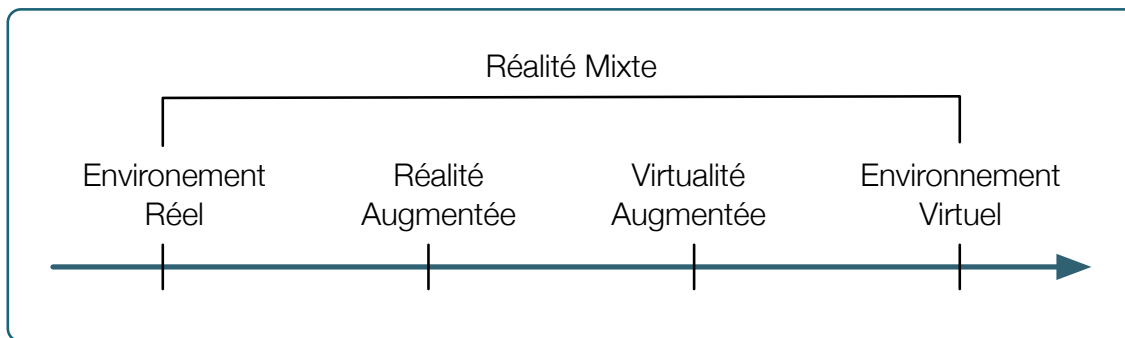


Figure 2: Le continuum réel-virtuel adapté de [Milgram and Kishino, 1994].

Dans ce chapitre, nous vous expliquons les multiples définitions et concepts de la réalité virtuelle. Puis, nous vous présentons les quatre tâches de l'interaction 3D en réalité virtuelle ainsi que leurs techniques associées. Enfin, nous terminons ce chapitre par l'étude de certains travaux de recherches concernant les architectures de systèmes multi-capteurs en réalité virtuelle.

1.2 LA RÉALITÉ VIRTUELLE

Une des premières choses à connaître sur la *réalité virtuelle* est que ce terme est une mauvaise traduction de *virtual reality*. En effet dans la langue française *réalité* et *virtuelle* sont deux termes ayant des sens opposés. *Réalité virtuelle* est donc un oxymoron, c'est-à-dire une figure alliant deux mots de sens contradictoire. *Réalité* dans la langue française désigne le monde dans sa globalité et la vérité. Alors que le terme *virtuel* signifie : potentiel. On parle donc de réalité potentielle. Cette mauvaise traduction démontre bien la difficulté à définir ce domaine. Plusieurs chercheurs ont proposé leur propre définition de la réalité virtuelle et nous allons vous en présenter quelques-unes.

Pour Tisseau, afin de définir la réalité virtuelle, il faut se référer à la notion du virtuel des sciences physiques [Tisseau, 2001]. En sciences physiques, le virtuel est une notion appliquée à des modèles à des fins d'explications et de prédictions de phénomènes difficiles à représenter. Ces modèles virtuels ont plus vocation à servir d'analogies à des représentations mathématiques complexes, plutôt que d'être une représentation rigoureuse de la réalité [Omnès, 1995]. Tisseau en a conclu donc que pour les physiciens, le virtuel sert d'outil d'intelligibilité du réel. Ce qui l'amène à dire :

"La réalité virtuelle est virtuelle parce qu'elle concerne résolument l'univers des modèles. Une réalité virtuelle est un univers de modèles au sein duquel tout se passe comme si les modèles étaient réels parce qu'ils proposent simultanément la triple médiation des sens, de l'action et de l'esprit."

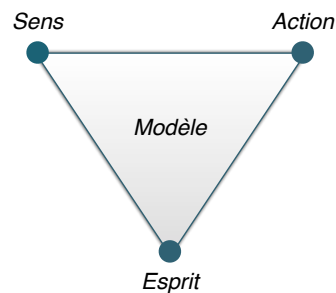


Figure 3: Les trois médiations du modèle en réalité virtuelle proposé par [Tisseau, 2001] Tisseau

Cette triple médiation est représentée figure 3. La médiation des sens permet à l'utilisateur d'observer le modèle à l'aide de ces sens. La médiation de l'action correspond à l'aptitude de l'utilisateur à tester en temps réel la réactivité du modèle. Quant à la médiation de l'esprit, elle définit la construction de la proactivité des modèles, en offrant la possibilité à l'utilisateur de les adapter à ses projets.

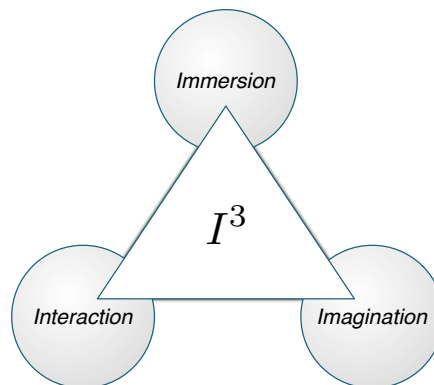


Figure 4: Les trois I de Burdea and Coiffet [Burdea and Coiffet, 1993].

Burdea and Coiffet ont, quant à eux, défini la réalité virtuelle autour de trois composantes qui sont l'Immersion, l'Interaction et l'Imagination [Burdea and Coiffet, 1993]. Ainsi, pour les auteurs, la réalité virtuelle est la présence d'un utilisateur en immersion au sein d'un environnement virtuel, autant réaliste que possible, dans lequel il peut interagir en temps réel.

Fuchs and Moreau ont proposé, dans “*Le traité de la réalité virtuelle : Volume 1*” [Fuchs and Moreau, 2003] une définition de la finalité de la réalité virtuelle:

“La finalité de la réalité virtuelle est de permettre à une personne (ou à plusieurs) une activité sensori-motrice et cognitive dans un monde artificiel, créé numériquement, qui peut être imaginaire, symbolique ou une simulation de certains aspects du monde réel. D’un point de vue physique, l’utilisateur est relié à une ou plusieurs machines au niveau de ses sens et ses réponses motrices via des interfaces motrices et sensorielles.”

On parle ici d’immersion et d’interaction (I^2) sensori-motrices [Fuchs and Moreau, 2003]. L’immersion se réalise grâce aux sens et l’interaction est effective grâce aux réponses motrices et aux sens de l’utilisateur.

Enfin, il faut savoir que la réalité virtuelle touche deux grands domaines:

- Le domaine des sciences et techniques à travers l’informatique, la robotique, la téléopération, l’optique, la mécanique, etc.
- Le domaine des sciences humaines à travers la psychologie, le comportement humain, l’ergonomie et la physiologie.

Tisseau a défini qu’une application de réalité virtuelle est une fonction définie par trois composantes [Tisseau, 2004] : *autonomie*, *interaction* et *immersion*. Une quatrième composante existe, la *présence*, sous la forme de la combinaison de l’*immersion* et de l’*interaction*.

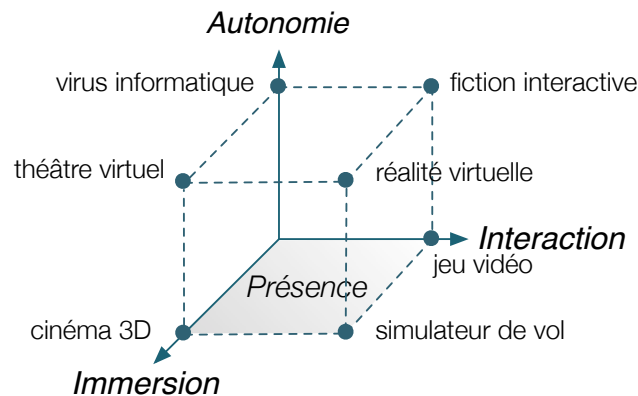


Figure 5: La présence et l'autonomie en réalité virtuelle, proposé par Tisseau [Tisseau, 2004].

Un modèle général pour la conception d’un dispositif de réalité virtuelle a été proposé par Fuchs and Moreau dans “*Le traité de la réalité virtuelle : Volume 1*” [Fuchs and Moreau, 2003] (figure 6). Ce modèle définit trois niveaux d’immersion et d’interaction:

- Au niveau d’ I^2 *Sensori-motrices* sont définis les liaisons physiques entre la machine et l’Homme.
- Au niveau d’ I^2 *Cognitives* sont définis les processus cognitifs qui rentrent en jeu et le logiciel de RV. L’ensemble doit permettre de rendre invisible à l’utilisateur le niveau I^2 Sensori-motrices.
- Au niveau d’ I^2 *Fonctionnelles* sont définis les activités de l’Homme dans l’environnement virtuel (EV). Ce niveau traite donc de l’application en elle-même et de ses objectifs.
- Un quatrième niveau peut être défini comme niveau d’ I^2 *Sociales* qui introduit l’aspect de travail collaboratif ou de téléconférence RV.

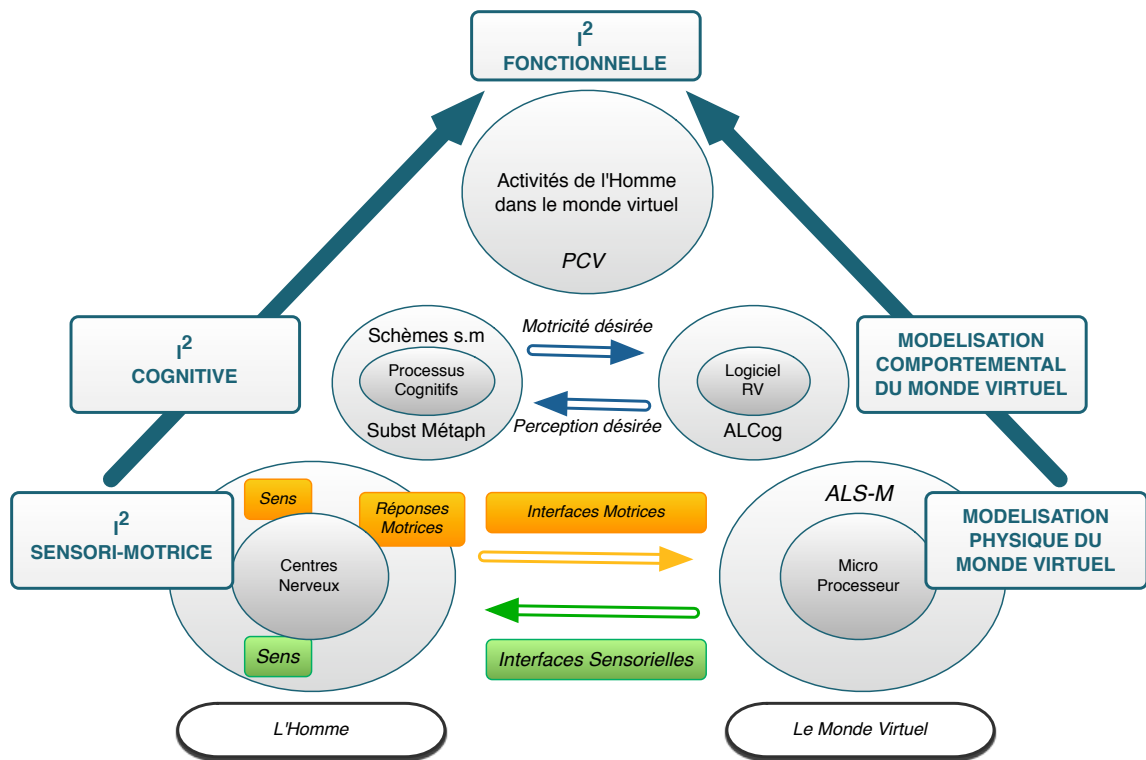


Figure 6: Schéma de Fuchs and Moreau représentant les 3 niveaux d'interaction en réalité virtuelle [Fuchs and Moreau, 2003].

1.3 L'INTERACTION 3D

1.3.1 Présentation

Dans un monde virtuel, l'interaction 3D peut être définie comme le pouvoir de naviguer, sélectionner, toucher et/ou manipuler des objets virtuels au sein d'un environnement virtuel. En plus de ces trois tâches de base, le système doit être capable de gérer et contrôler ces différentes tâches [Bowman, 2004]. Si par ailleurs l'utilisateur ne peut pas interagir efficacement avec des objets appartenant à l'Environnement Virtuel (EV), alors simplement d'autres tâches à niveau plus élevé ne peuvent pas être accomplies. Dans les sections suivantes, nous présentons les différentes tâches d'interaction 3D ainsi que leurs techniques d'interaction 3D courantes associées. Un état de l'art plus complet sur les techniques d'interaction 3D en réalité virtuelle est présenté dans [Ouramdane et al, 2009].

1.3.2 Trèfle fonctionnel de l'interaction 3D

Comme nous avons vu précédemment, il existe quatre tâches d'interaction 3D. Cependant, le contrôle d'application est une tâche système présente à tous les niveaux de l'interaction 3D. La navigation, la sélection et la manipulation sont les trois tâches fonctionnelles de l'utilisateur ; elles sont représentées par un trèfle fonctionnel de l'interaction 3D (figure 7) comportant trois espaces fonctionnels :

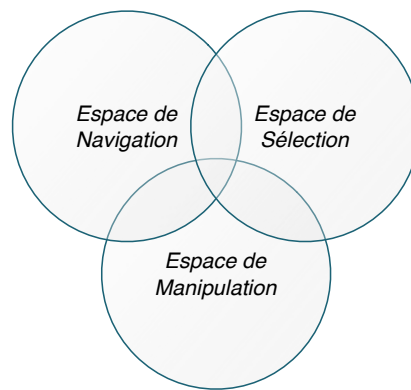


Figure 7: Trèfle fonctionnel de l'interaction 3D extrait de [Otmane et al, 2007].

- Espace de navigation (EN) : Cet espace regroupe l'ensemble des fonctionnalités nécessaires à l'utilisateur pour modifier sa position et son orientation dans l'environnement virtuel. La navigation des autres utilisateurs est aussi prise en compte dans cet espace-là, c'est ce qu'on appelle la conscience de la navigation.
- Espace de sélection (ES) : Cet espace regroupe l'ensemble des fonctionnalités nécessaires à l'utilisateur pour sélectionner un objet ou un ensemble d'objets virtuels dans l'environnement virtuel. Chaque utilisateur a aussi la possibilité d'avoir une vue d'ensemble des sélections effectuées par les autres utilisateurs dans l'environnement virtuel, c'est ce qu'on appelle la conscience de sélection.
- Espace de manipulation (EM) : Cet espace regroupe l'ensemble des fonctionnalités nécessaires à l'utilisateur pour modifier les propriétés d'un objet ou d'un ensemble d'objets, ces propriétés peuvent être la position, l'orientation, la couleur, la taille, etc. Les manipulations des autres utilisateurs sont aussi prises en compte dans l'environnement virtuel, c'est ce qu'on appelle la conscience de manipulation.

1.3.3 Les tâches de l'interaction 3D en réalité virtuelle

1.3.3.1 La Tâche de navigation

La navigation est la tâche la plus commune à tous les EVs. On distingue deux composantes principales à cette tâche :

- Le *déplacement* (*Travel*), qui représente la composante mobile de cette tâche, elle se rapporte au déplacement physique d'un endroit à un autre.
- La *recherche d'itinéraire* (*Wayfinding*), qui représente la composante cognitive c'est-à-dire la manière dont on va se déplacer jusqu'à l'emplacement voulu. Elle permet aux utilisateurs de se repérer dans l'environnement et de choisir une trajectoire pour se déplacer [Fuchs and Moreau, 2003]. L'utilisateur se pose des questions telles que : «où suis-je?», «où doit-je aller?», «comment arrivé là?», etc.

Cette tâche de la navigation permet à l'utilisateur de réaliser trois tâches principales, qui sont :

- L'*exploration*, c'est-à-dire la navigation au sein de l'environnement virtuel sans but explicite. Il s'agit pour l'utilisateur de découvrir et de connaître l'environnement.
- La *recherche*, c'est-à-dire naviguer vers une cible dont la position est, ou n'est pas, connue.

- La *Manœuvre*, c'est-à-dire réaliser des mouvements courts et précis dans le but de réaliser une tâche demandant de l'agilité.

1.3.3.2 La tâche de sélection

La sélection représente la désignation d'un objet ou d'un ensemble d'objets pour un objectif donné. Cette tâche demande deux étapes : la désignation d'un objet pouvant être sélectionné et l'étape de sélection même de l'objet. L'évènement de sélection peut être indiqué de plusieurs façons telles que : presser un bouton, utiliser un geste ou une commande vocale, mais il peut être réalisé automatiquement si le système d'interaction prend en considération les intentions de l'utilisateur.

1.3.3.3 La tâche de manipulation

La manipulation se rapporte à la spécification et la modification des propriétés des objets. C'est la composante la plus active de l'interaction 3D puisque c'est la tâche dans laquelle l'utilisateur interagit directement avec l'environnement virtuel. Cette tâche est intimement liée à la sélection, car on ne peut pas manipuler un objet sans l'avoir sélectionné avant.

Du point de vue des transformations géométriques, on peut subdiviser la tâche de manipulation en trois sous-tâches :

- Le positionnement, qui consiste à modifier la position d'un objet.
- La rotation, qui permet de changer l'orientation d'un objet.
- La mise à l'échelle, qui joue sur le facteur d'échelle d'un objet suivant les trois axes, engendrant ainsi un rétrécissement, un agrandissement ou une homothétie.

Cependant, d'autres paramètres peuvent être modifiés, tels que la couleur, la texture, le maillage de l'objet et bien d'autres. On distingue deux types de manipulation en réalité virtuelle :

- La manipulation directe, la plus naturelle, puisque l'utilisateur agit directement sur l'objet afin d'appliquer les modifications souhaitées.
- La manipulation indirecte, dans laquelle l'utilisateur modifie les propriétés de l'objet au travers d'entités graphiques 2D (menus, boutons, etc) ou 3D (outils virtuels, widgets, etc).

Au moment du passage de la tâche de sélection à la tâche de manipulation, plusieurs questions peuvent être posées :

- Comment une technique de manipulation peut-elle s'intégrer facilement avec une technique de sélection déjà choisie et utilisée?
- Que se passe-t-il lorsque l'objet sélectionné est relâché?

Pour résoudre ces problèmes, il faut d'une part désactiver la technique de sélection utilisée ainsi que les retours d'informations correspondants et d'autre part, activer la technique de manipulation adéquate selon les conditions d'une application donnée.

1.3.3.4 Le système de commande / contrôle d'application

Cette tâche permet le changement d'état du système ou du mode d'interaction 3D.

1.3.4 Lien entre les tâches de l'interaction 3D

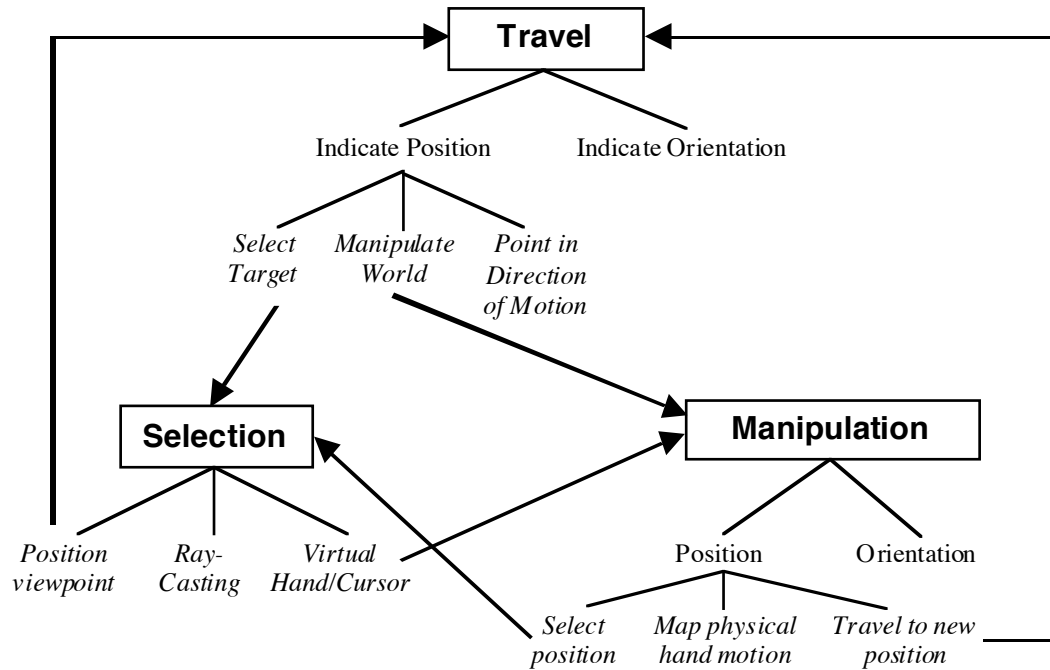


Figure 8: Lien entre les tâches de l'interaction 3D définis par Bowman [Bowman, 1999].

Bowman, dans sa thèse [Bowman, 1999], a souligné que certaines techniques de l'interaction 3D pouvaient réaliser d'autres tâches que celles pour laquelle elles avaient été conçues avec seulement une ou quelques modifications. Le principal intérêt d'avoir une technique multitâche de l'interaction 3D est d'utiliser la même métaphore et donc d'augmenter d'une part la cohérence de l'interface et d'autre part, de réduire la charge cognitive de l'utilisateur, puisqu'il n'a plus besoin de se familiariser avec plusieurs métaphores. Bowman en a donc conclu que les taxonomies de chacune des trois tâches (navigation, sélection et manipulation) étaient liées. Il a donc proposé le schéma représenté dans la figure 8, montrant les liens entre les différentes tâches.

1.3.5 Les techniques d'interaction 3D courantes en réalité virtuelle

Il existe un nombre considérable de techniques pour interagir en 3D dans un environnement virtuel. Par technique d'interaction 3D, on désigne la méthode utilisée pour accomplir une des tâches de l'interaction 3D (navigation, sélection, manipulation). Mine [Mine, 1995] a classifié toutes ces techniques d'interaction 3D en trois catégories:

- *L'interaction directe par l'utilisateur* qui inclut toutes les techniques qui utilisent le tracking de la tête, de la main, la reconnaissance de gestes, etc, afin de transformer une action réelle de l'utilisateur en action virtuelle dans l'environnement virtuel.
- Les *contrôleurs physiques*, tels que les boutons, les manettes de jeux, les claviers, les volants, les joysticks, etc. au travers desquels l'utilisateur interagit dans l'environnement virtuel. Le principal avantage de ce mode d'interaction, est qu'il est généralement plus aisé pour l'utilisateur d'interagir avec précision sur l'environnement virtuel. Cependant cet avantage est aussi la faiblesse de ce mode, car suivant l'interface matérielle utilisée, l'interaction n'est plus naturelle et demande un apprentissage.
- Les *contrôleurs virtuels*, qui représentent en quelque sorte les contrôleurs physiques mais modélisés dans l'environnement virtuel. Cependant, virtualité aidant, on peut même inventer des contrôleurs n'existant pas dans la réalité. L'imagination est la seule limite. Néanmoins, il faut réfléchir à leurs modélisations, ainsi que la facilité avec laquelle l'utilisateur pourra les utiliser, sous peine de pénaliser grandement l'interaction 3D avec ceux-ci.

À chacune des tâches élémentaires de l'interaction 3D correspond des techniques qui permettent d'effectuer la tâche désirée. Voici, une présentation non exhaustive des techniques d'interaction 3D courantes en fonction de la tâche de l'interaction 3D. Une présentation plus complète peut être lue dans la revue : [Ouramdane et al, 2009].

1.3.5.1 Les techniques de navigation

Les techniques de navigation permettent à l'utilisateur de se mouvoir dans l'environnement virtuel. Même s'il existe différents moyens de se déplacer, le mouvement de l'utilisateur dans le monde virtuel est défini par seulement deux paramètres qui sont : la direction du mouvement et la vitesse de déplacement. Ces deux paramètres peuvent être regroupés en un seul paramètre qui est le vecteur vitesse.

Il existe également plusieurs approches pour se déplacer dans un environnement virtuel. La manière la plus naturelle, sera de transformer directement le mouvement de l'utilisateur dans le monde réel en mouvement dans le monde virtuel. Cependant, on atteint vite la limite de cette approche dès lors que le monde virtuel est beaucoup plus vaste que ce que permet le monde réel autour de l'utilisateur. Ceci, soit pour des raisons de technologies (utilisateur hors de portée des capteurs) ou tout simplement pour des contraintes physiques (pièce trop petite ou obstacle(s)). C'est pour cette raison que d'autres approches existent, mais avec celles-ci le déplacement se fait de manière indirecte, ou relative. On utilise la métaphore de la téléportation, ou du vol libre ou de la conduite d'un véhicule.

Parmi les techniques de navigation utilisant, la direction du mouvement de l'utilisateur, et en particulier de sa tête, on retrouve la technique *Direction du regard* (*Gaze directed steering*) [Mine, 1995]. Cette technique consiste à déplacer le point de vue dans la direction du regard, elle-même déterminée par les mouvements de la tête (tracking de la tête). Cette technique ne permet pas à l'utilisateur de voir sur les côtés en se déplaçant. De plus elle n'est efficace que dans des environnements virtuels immersifs.

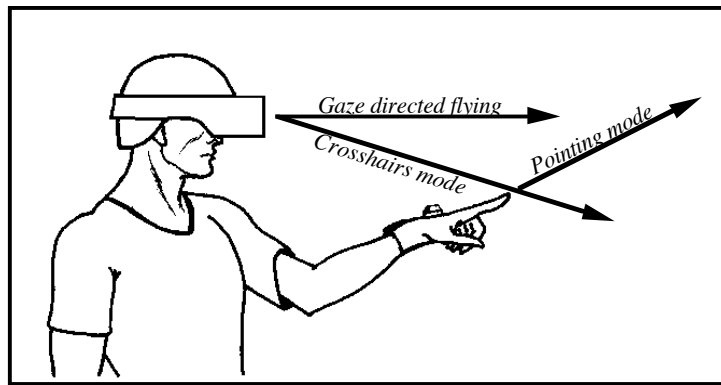


Figure 9: La technique direction du regard [Mine, 1995].

Une variante de la technique *Direction du regard* existe sous la forme de la technique *Mode orbital*, dans laquelle l'objet est toujours positionné face à la tête de l'utilisateur cependant son orientation dans le référentiel du monde reste la même. Ceci a pour effet de produire une impression de déplacement autour de l'objet, lorsque l'utilisateur oriente sa tête [Chung, 1993].

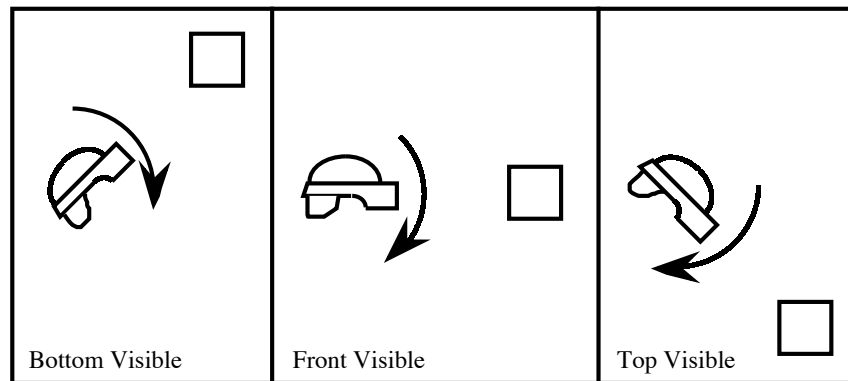


Figure 10: Fonctionnement du mode orbital [Mine, 1995].

Cependant, il n'y a pas que la tête qui peut être utilisée pour indiquer la direction dans laquelle on veut se déplacer dans l'environnement virtuel, la main peut également être employée. Ainsi avec la technique *Direction de la main (Pointing)* [Mine, 1995], l'utilisateur spécifie constamment la direction du mouvement en utilisant sa main. Dans cette technique la main de l'utilisateur est traquée. Elle permet de se déplacer et de regarder dans différentes directions. Cependant cela nécessite que l'utilisateur comprenne comment la position et l'orientation de sa main influe sur le sens du déplacement.

Dans la technique *Déplacement en utilisant une carte (Goal driven)*, l'utilisateur déplace une icône, le représentant, sur une carte 2D par la pointe d'un stylet ou du doigt jusqu'à un nouvel endroit sur la carte, ce qui implique le déplacement de l'utilisateur dans l'environnement virtuel.

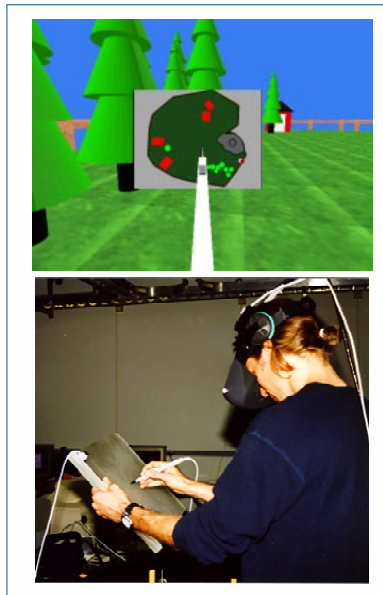


Figure 11: Une personne entrain d'utiliser la technique : *Déplacement sur une carte* (photo issue de [Bowman, 1999]).

Dans ce contexte, on peut également citer la technique *Magallanes* [Abásolo and Della, 2007], qui consiste en la génération d'une interface graphique attachée à la scène qui facilite la sélection de points d'intérêt. Lorsque l'utilisateur choisi une destination, la technique trouve automatiquement le chemin optimal partant de l'avatar.



Figure 12: Image illustrant la technique *Magallanes* avec l'interface graphique attachée à la scène [Abásolo and Della, 2007].

La technique d'interaction 3D *Déplacement en utilisant les gestes de la main*, proposée par [Mapes and Moshell \[Mapes and Moshell, 1995\]](#), consiste à utiliser les gestes de la main pour se déplacer dans l'environnement virtuel. La technique *déplacement en utilisant les gestes de la main* est basée sur la métaphore de tirer sur une corde.

Afin de régler la vitesse de déplacement, [Mine](#) a répertorié les différentes options. Ainsi, imposer une vitesse constante est la manière la plus simple de répondre à ce problème. Cependant elle nécessite de régler sa valeur en fonction de l'échelle du monde virtuel sous peine de provoquer un déplacement trop lent ou trop rapide. De plus, l'utilisateur aura, avec cette option, tendance à "louper" les endroits où il est censé s'arrêter.

Une autre approche consiste à utiliser une accélération constante. Dans ce cas-là, l'utilisateur commence à se déplacer avec une vitesse lente puis l'accélération constante agissant, la vitesse augmente continuellement. Là encore, un ajustement de ce paramètre est nécessaire en fonction de la taille de l'environnement virtuel.

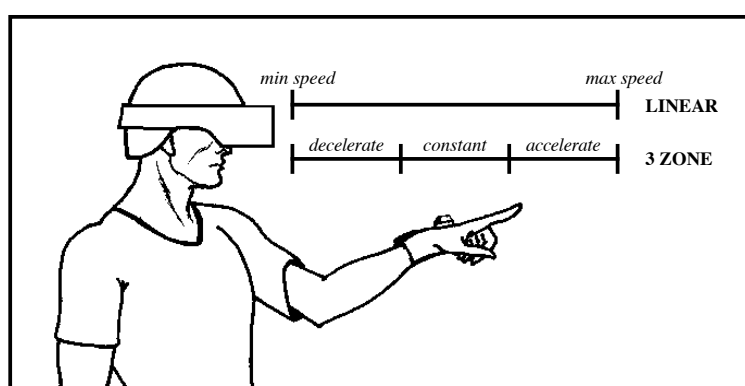


Figure 13: Contrôle de la vitesse par la main [[Mine, 1995](#)].

L'approche offrant le plus d'adaptabilité est celle utilisant la main pour contrôler la vitesse de déplacement. En jouant sur la position de sa main, l'utilisateur peut spécifier la vitesse et l'accélération de déplacement. Différentes transformations peuvent être définies, c'est le grand avantage de cette approche. Un aperçu de cette méthode est représenté figure 13.

Parmi les techniques de navigations plus récentes, on peut citer les travaux de [[Tan et al, 2001](#)] qui décrivent la technique : *Speed-coupled Flying with Orbiting*. Cette technique permet un contrôle de la vitesse de déplacement de la caméra virtuelle en fonction de la hauteur et l'inclinaison de celle-ci. Cela afin de permettre de passer sans problème d'un point de vue local à un point de vue global de l'environnement virtuel. L'utilisateur a aussi la possibilité de se placer en orbite autour d'un objet spécifique pour pouvoir l'observer.

On peut également citer une autre technique récente, appelée *Navidget* [[Hachet et al, 2009](#)] propose d'utiliser un widget d'interaction permettant de positionner par le biais d'une interaction 2D facilement une ou plusieurs caméras virtuelles dans un environnement 3D. Cette technique est elle-même basée sur la technique *Point Of Interest* proposée par [[Mackinlay et al, 1990](#)].

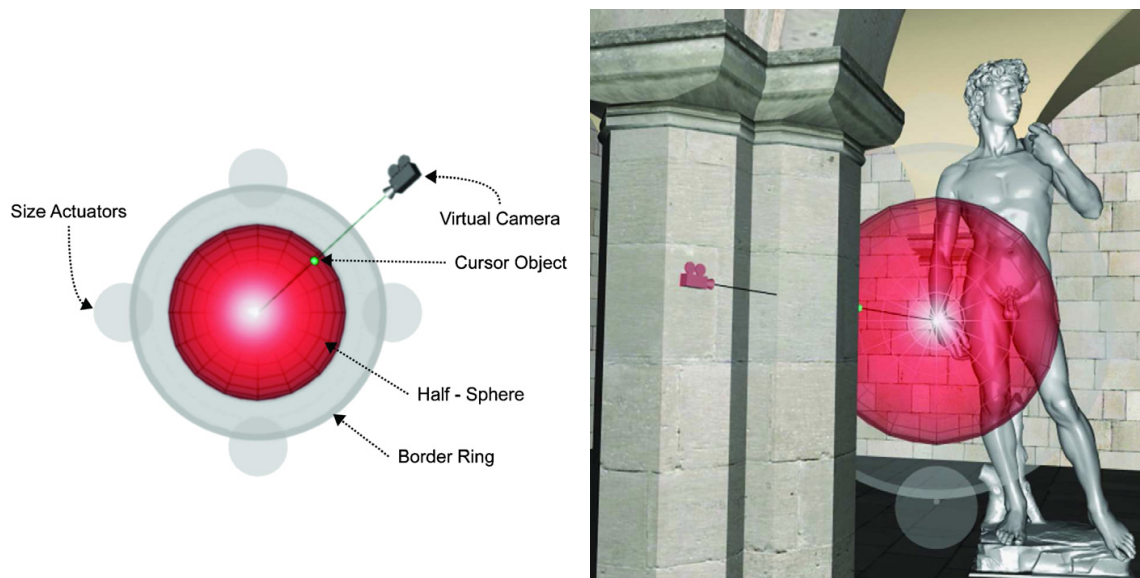


Figure 14: Le Navidget (à gauche) et le Navidget en utilisation dans une application de réalité virtuelle (à droite) [Hachet et al, 2009].

Enfin, des contrôleurs physiques ou virtuels peuvent être utilisés pour contrôler la vitesse de déplacement. Ces deux approches ont les mêmes avantages et inconvénients que ceux présentés dans la section 1.3.5.

1.3.5.2 Les techniques de sélection

Mine a proposé deux catégories de techniques de sélection en réalité virtuelle [Mine, 1995]. La première : la sélection locale, est la plus naturelle, car elle copie ce qu'il se passe dans le monde réel, c'est-à-dire la sélection d'un objet proche de l'utilisateur dans l'environnement virtuel. L'inconvénient principal de ces techniques provient justement du constat que l'utilisateur ne peut sélectionner que les objets qui lui sont proches. La seconde est la sélection à distance, qui même si elle est moins naturelle pour l'utilisateur, est souvent plus pratique puisque ces techniques permettent de s'affranchir de se déplacer jusqu'à l'objet pour le sélectionner.

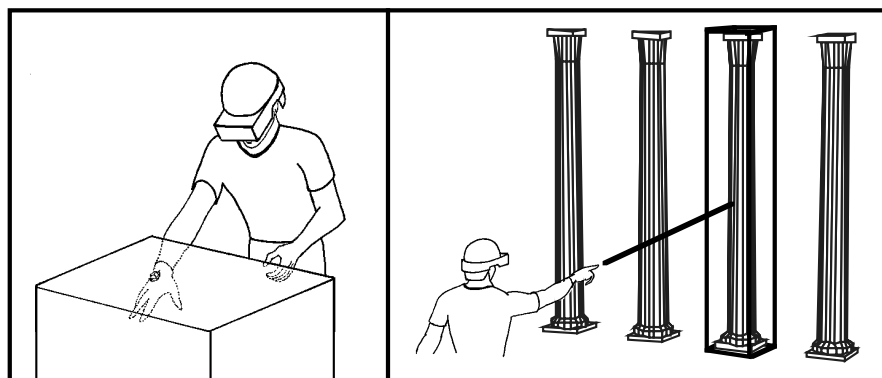


Figure 15: Comparaison des approches des techniques de sélection locale et à distance [Mine, 1995].

Parmi les techniques de sélection locales on retrouve la technique de la *Main virtuelle simple* (*Virtual hand*) Cette technique [Bowman, 2004], représentée figure 16, s'inspire directement du schème de la sélection des objets par l'être humain dans le monde réel. Par cette technique, l'objet peut être sélectionné en le touchant ou en détectant l'intersection de la main virtuelle avec l'objet. La technique *Main virtuelle simple* est caractérisée par une correspondance parfaite entre la main virtuelle et la main réelle (c'est-à-dire les mouvements de la main réelle sont directement représentés par la main virtuelle).

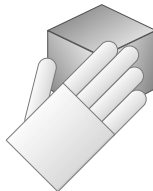


Figure 16: La technique d'interaction 3D de la *Main virtuelle simple*.

Cependant, le principe même de la réalité virtuelle autorise l'utilisateur à agir d'une manière qu'il ne pourrait pas réaliser dans le monde réel, la majorité des techniques de sélection sont donc à distance, en voici un échantillon des plus courantes. Jacoby et al ont proposé une technique dite du *Rayon laser virtuel* (*Raycasting*) [Jacoby et al, 1996]. Cette technique, représentée figure 17, est basée sur la métaphore du pointeur virtuel. Un rayon laser infini part de la main virtuelle et traverse tout le monde virtuel. Le premier objet en intersection avec le rayon laser peut être sélectionné.

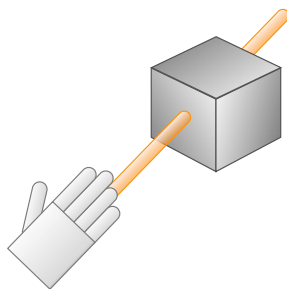


Figure 17: La technique d'interaction 3D du *Raycasting*.

La technique *Occlusion* a été conçue par Pierce et al [Pierce et al, 1999]. Cette technique, représentée figure 18, fonctionne dans le plan image (2D), en couvrant l'objet désiré avec l'objet sélectionneur (doigt). L'objet le plus proche du rayon visuel émanant du doigt peut être sélectionné.

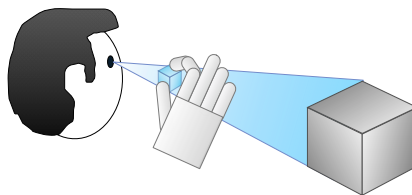


Figure 18: La technique d'interaction 3D *Occlusion*.

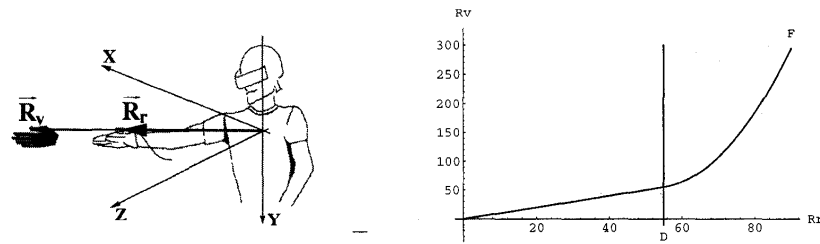


Figure 19: Représentation de la technique Go-Go et sa fonction mathématique définissant l'extension de la main virtuelle [Poupyrev et al, 1996].

La technique Go-Go proposée par Poupyrev et al [Poupyrev et al, 1996] est aussi appelée technique d'extension du bras. Elle est basée sur la métaphore de la *Main virtuelle*. Comme dans la technique de la *Main virtuelle simple*, on touche les objets pour les sélectionner. La relation entre la main réelle et virtuelle est non linéaire. En fait, à partir d'un certain seuil D, la main virtuelle part en direction de l'objet.

La *commande vocale* peut également être utilisée pour sélectionner un objet, en énonçant son identifiant par exemple, mais l'utilisateur peut aussi sélectionner un objet à partir d'une liste contenant tous les noms des objets.

Une classification de toutes ces techniques a été réalisée par Bowman [Bowman, 1999], on peut l'apercevoir dans la figure 20.

Techniques de sélection

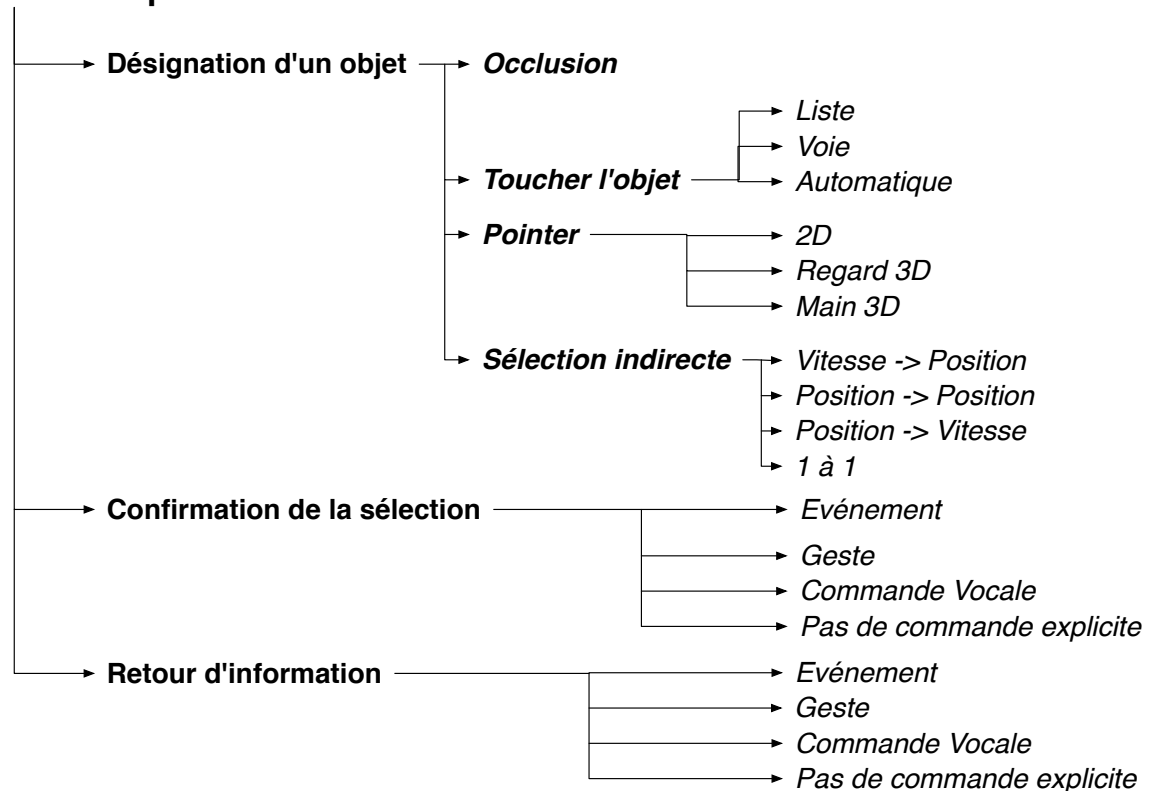


Figure 20: Classification par décomposition des tâches des techniques de sélection extrait de [Bowman, 1999].

1.3.5.3 Les techniques de manipulation

Les techniques de manipulation ont pour but de modifier les propriétés de l'objet sélectionné. Les principaux paramètres qui peuvent être modifiés sont la position, l'orientation, le point de pivot (autour duquel les rotations de l'objet ont lieu) ainsi que le facteur d'échelle de l'objet.

Ces techniques sont souvent associées ou bien dérivées des techniques de sélection. Une des techniques qui évoque le plus ce rapprochement est la technique de *Main virtuelle simple*. Cette technique [Bowman, 2004] est utilisée à la fois pour les tâches de sélection et de manipulation. L'application de cette technique à la manipulation nécessite des modifications dans le graphe de scènes en attachant l'objet sélectionné à la main virtuelle. Dans un premier temps, on attache l'objet à la main pour le manipuler (l'objet hérite du mouvement de la main). Dans un second temps on re-attache l'objet au monde après la manipulation.

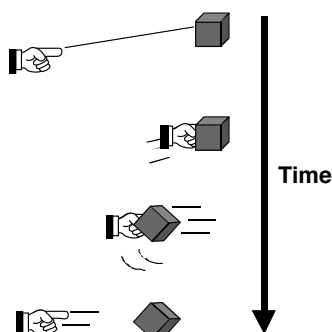


Figure 21: Chronologie de l'utilisation de la technique HOMER [Bowman, 1999].

Un autre exemple de la relation qui existe entre les techniques de sélection et de manipulation, est la technique *HOMER* [Bowman and Hodges, 1997; Bowman, 1999]. Cette technique est une technique hybride, elle combine la technique *Raycasting* pour la sélection et la technique *Main virtuelle simple* pour la manipulation. Après la sélection d'un objet à l'aide du *Raycasting*, la main virtuelle se déplace automatiquement jusqu'à l'objet qui sera attaché à cette dernière. Une fois que la manipulation est terminée, la main virtuelle reprend sa position initiale et l'objet manipulé garde sa position actuelle. On peut également noter le travail de Wilkes and Bowman qui proposent une version améliorée de la technique *HOMER* en lui ajoutant un gain dynamique en fonction de la vitesse de déplacement [Wilkes and Bowman, 2008].

Le *changement d'échelle* [Stoakley et al, 1995] est une technique hybride. Elle est souvent associée à la technique de sélection par *occlusion*. Lorsque l'objet est sélectionné dans le plan image, la représentation de l'utilisateur dans l'EV est agrandie ou l'objet virtuel est réduit afin que la main virtuelle touche vraiment l'objet occulté.

La technique *Monde en miniature*, conçue par Pausch et al [Pausch et al, 1995], représentée figure 22, utilise une représentation miniature du monde virtuel pour permettre à l'utilisateur de faire des manipulations indirectes sur les objets du monde virtuel. La manipulation est faite avec la technique *main virtuelle simple*.

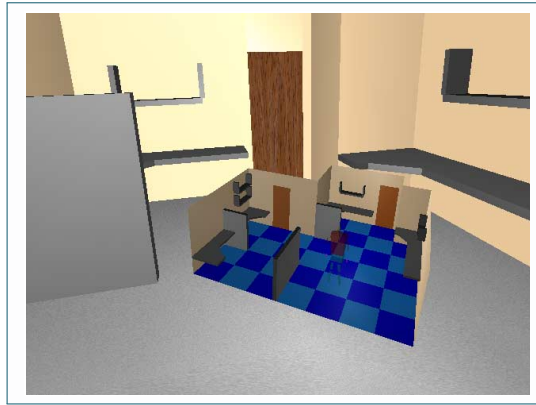


Figure 22: Aperçu de la technique *Monde en miniature*[Pausch et al, 1995].

Le principe de cette technique est le suivant : le monde virtuel est tenu dans la main de l'utilisateur afin qu'il puisse réaliser des manipulations sur celui-ci indirectement. Toute action sur un objet dans le monde virtuel miniature provoque la même action dans le monde virtuel initial.

Ouramdane et al ont développé une technique d'interaction 3D, appelée *Follow-me*, afin de pallier le manque de précision des techniques *Go-Go* et *Raycasting* lorsque l'objet cible est distant ou petit. *Follow-me* peut être vue comme un complément à ces deux techniques [Ouramdane et al, 2006b]. Le principe est le suivant : l'EV est divisé en trois zones dans lesquelles l'interaction 3D possède sa propre granularité.

- La première zone est appelée *zone de manipulation libre*. Dans cette zone, la granularité d'interaction est importante. Un mouvement dans le monde réel est retranscrit dans le monde virtuel avec un gain égal ou supérieur à un.
- La seconde zone est appelée *zone de manipulation mise à l'échelle*. Dans cette zone, la granularité d'interaction est moyenne. Un mouvement ample dans le monde réel est retranscrit comme un mouvement moins ample dans le monde virtuel avec un gain compris entre zéro et un.
- La dernière zone est appelée *zone de manipulation précise*. Dans cette zone la granularité d'interaction est faible. On utilise des guides virtuels pour assister l'utilisateur et il n'y a plus qu'un seul degré de liberté. Les mouvements de l'utilisateur seront donc interprétés comme avancer pour s'approcher de la cible ou reculer pour s'éloigner de la cible.

Dans un premier temps, la technique *Follow-me* a été testée pour la tâche de sélection. Les résultats des évaluations qui ont été faites montrent que cette technique est plus efficace que la technique *Go-Go* pour la sélection des objets petits et/ou lointains. Par rapport à la technique *Raycasting*, *Follow-me* n'améliore pas beaucoup les performances dans le cas des objets statiques. Dans le cas des objets dynamiques, *Follow-me* est nettement plus efficace que les techniques *Go-Go* et *Raycasting*.

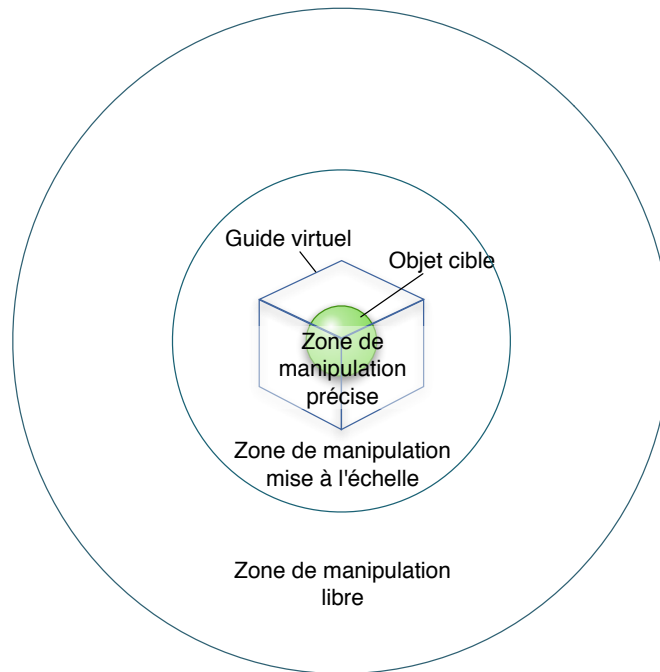


Figure 23: Zones d'interaction de la technique Follow-me.

1.3.5.4 Menus graphiques et widgets

A - MENUS 3D ORIENTÉS PAR LA MAIN

Ces menus consistent à afficher un objet circulaire contenant des éléments à sélectionner. Après initialisation, l'utilisateur tourne sa main jusqu'à ce que l'élément désiré tombe dans la zone de sélection.

B - MENUS 2D TRANSFORMÉS

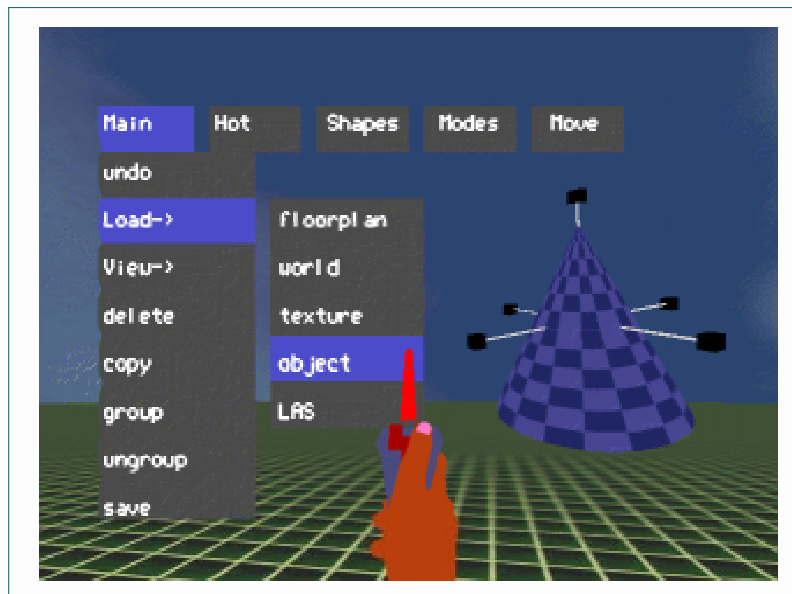


Figure 24: Un exemple de menu 2D transformés afin de les rendre compatibles dans un espace 3D [Bowman, 1999].

Ces menus sont tout simplement une variante des menus que l'on retrouve dans les bureaux 2D des systèmes d'exploitation mais avec plus de degrés de liberté.

C - WIDGETS 3D

Les widgets sont des éléments graphiques d'interface. Dans les EV, les widgets 3D sont utilisés pour changer les fonctionnalités du système de commande de l'environnement ou de l'objet sélectionné. Le choix de l'emplacement des widgets 3D dans l'espace est primordial car c'est un menu extrêmement sensible au contexte.

Un exemple récent de widget 3D peut être observé dans le logiciel *3D's Max*. Proposé par Khan et al, le *ViewCube* [Khan et al, 2008], représenté figure 25 est un widget servant à la fois d'indicateur d'orientation ainsi qu'un contrôleur du point de vue. Il permet à un utilisateur de comprendre quelle est l'orientation de la caméra virtuelle qui scrute la scène ainsi que de choisir aisément un autre point de vue en cliquant directement sur les faces, arrêtes et coins du widget.

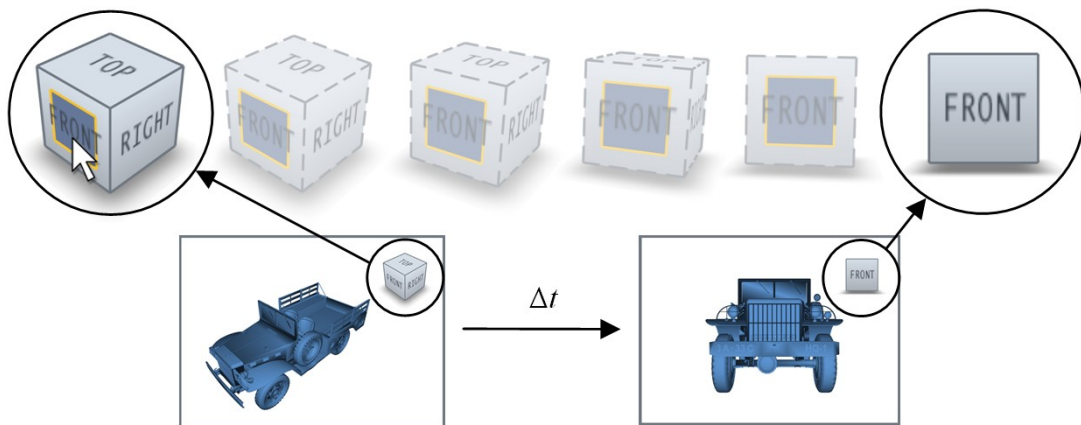


Figure 25: Le *ViewCube* : en cliquant sur la face avant du cube, le widget (cube) ainsi que la scène 3D s'orientent de manière à ce disposer en vue de face [Khan et al, 2008].

1.3.5.5 Commande vocale

Cette technique peut être utilisée pour sélectionner, initialiser ou commander. En utilisant cette technique, il faut penser à pouvoir l'activer ou la désactiver afin d'éviter les erreurs d'interprétation.

1.3.5.6 Interaction gestuelle

Cette technique peut être utilisée pour initialiser, sélectionner ou commander. Elle utilise la main comme outil d'interaction.

1.3.5.7 Outils

A - OUTILS PHYSIQUES

Les outils physiques peuvent servir à réaliser une ou plusieurs fonctions. Le système de commande est directement intégré dans le dispositif.

B - OUTILS VIRTUELS

Les outils virtuels utilisent les signaux proprioceptifs pour placer le système de commande autour du corps de l'utilisateur. Ces outils virtuels peuvent être ordonnés sous la forme d'un menu ou bien comme dans le monde réel sous la forme d'une boîte à outils.

1.4 LES ASSISTANCES À L'UTILISATEUR POUR L'INTERACTION 3D

De nombreux travaux de recherche sur l'interaction 3D se sont intéressés sur les assistances à l'utilisateur pour l'interaction 3D. Ces assistances ont pour fonction principale d'aider l'utilisateur à accomplir des tâches dans l'environnement virtuel. La difficulté posée par ses assistances est que si le système cherche trop à aider l'utilisateur et qu'il se trompe sur la volonté de l'utilisateur, il risque de nuire à l'utilisateur lors de la réalisation de la tâche. De plus, certains utilisateurs n'aiment pas être assistés et cela leur procure plus une gêne qu'autre chose. Une autre difficulté est que ces assistances se doivent d'être les plus transparentes possibles. Dans le cas où elles ne le sont pas, des informations doivent être envoyées à l'utilisateur de manière à ce qu'il puisse comprendre comment le système tente de l'aider afin que ces assistances soient bien perçues par l'utilisateur final. On trouve différentes sortes d'assistances à l'interaction 3D, nous allons vous en présenter quelques-unes.

1.4.1 L'ontologie

L'ontologie définit la modélisation d'un ensemble de connaissances dans un domaine donné. Dans le cas de la réalité virtuelle, l'ontologie est généralement une base de données regroupant la liste des objets virtuels, leurs attributs et leurs classes. Mais on peut également décrire la liste des tâches à réaliser, la relation entre les différents objets etc. La plupart des systèmes d'interaction 3D multimodale sont basés sur le concept d'ontologie. Mais tout d'abord expliquons ce que c'est une ontologie. La définition donnée par le Petit Larousse est la suivante :

"Etude de l'être en tant qu'être, indépendamment de ses déterminations particulières."

Ce composant peut s'apparenter à une base de données de structure relativement complexe, qui permet à des ordinateurs de traiter son contenu. Cette base de données définit en réalité un jeu de règles, de relations et de concepts qui décrivent un domaine d'intérêt. Il existe deux types d'ontologie :

- *L'ontologie dépendante du domaine* : Cette ontologie est spécifique au domaine de l'application, on peut être amené à la redéfinir.
- *L'ontologie indépendante du domaine* : Cette ontologie est universelle, une fois définie, on peut la réutiliser dans n'importe quel domaine d'application.

L'inconvénient de ce type d'approche est qu'il est nécessaire de tout décrire dans cette base de données, ce qui peut être une tâche fastidieuse. De plus, l'environnement virtuel perd en flexibilité, puisque le système s'appuie grandement sur ce qui est décrit.

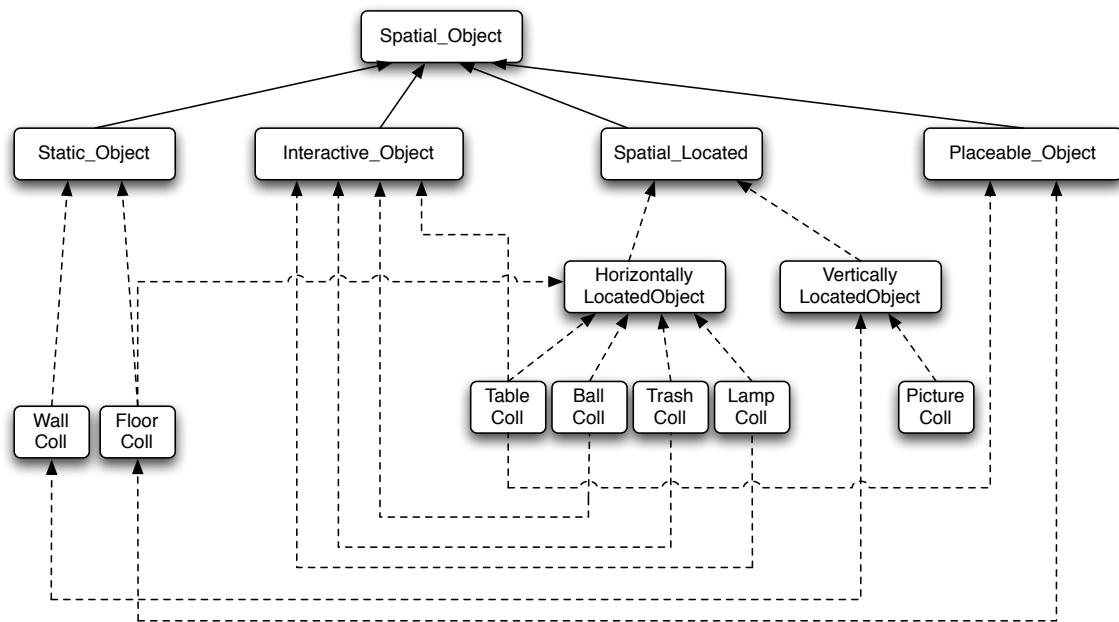


Figure 26: Exemple d'ontologie pour le placement spatial d'un objet [Irawati et al, 2005].

1.4.2 La conscience de l'interaction

Ce type d'assistance à l'interaction 3D a pour but d'essayer de se substituer à ce que nous réalisons constamment de manière inconsciente, par le biais de notre cerveau, c'est-à-dire l'étude du contexte à partir d'observation généralement visuelle. Ainsi on cherche à observer quels sont les objets visibles, cachés, ceux qui sont susceptibles d'attirer l'attention de l'utilisateur etc. Beaucoup de ce type d'assistance se retrouve dans les modèles d'interaction 3D collaborative. Mais nous nous contenterons juste de présenter quelques travaux de recherches dans ce domaine sans pour autant rentrer dans le vif du sujet.

Otmane et al ont proposés un modèle d'Interaction 3D collaborative [Otmane et al, 2007]. Ce modèle est basé sur les concepts de la conscience de groupe (Focus, nimbus et degrés d'intérêt). Ces concepts sont associés aux trois tâches de l'Interaction 3D (I3D) (la navigation, la sélection et la manipulation). Le modèle mis en place est un modèle générique permettant de gérer les interactions 3D entre les différents utilisateurs ainsi qu'entre les différents utilisateurs et les objets virtuels. Le système observe quels objets sont visibles par l'utilisateur et ceux qui sont susceptibles d'attirer son attention, etc. Ce modèle définit un certain nombre de concepts, qui sont traduits sous forme de matrices. Ces matrices peuvent être exploitées afin d'enrichir l'expérience de l'utilisateur, de l'assister afin de lui faciliter l'accomplissement de sa tâche ou bien d'interpréter ses intentions.

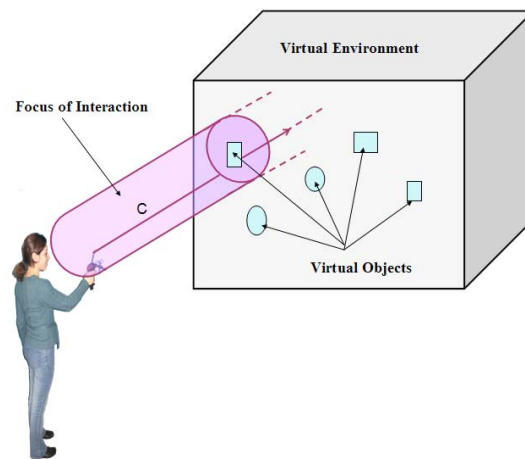


Figure 27: Un exemple de conscience de l'interaction, illustrant le focus de l'utilisateur, c'est-à-dire le volume dans lequel seront situés les objets qui seront susceptibles de l'intéresser [Otmane et al, 2007].

1.4.3 Les guides virtuels

Les guides virtuels sont des assistances à l'interaction 3D permettant à l'utilisateur de se placer précisément ou de placer un objet qu'il manipule à un emplacement précis. Généralement ils sont conçus soit autour d'une limitation des degrés de libertés, soit sur le suivi d'une trajectoire pré-calculée ou bien par un système de grille aimantée. Il faut savoir que les guides virtuels sont directement issus des logiciels de graphisme et de DAO - CAO dans lesquels ce type d'aide est présent depuis des années.

Ainsi, Otmane and Davesne ont utilisé des guides virtuels pour aider l'utilisateur à placer la pince d'un robot virtuel au-dessus de la pièce à saisir [Otmane and Davesne, 2009] (figure 28 et ainsi permettre à l'utilisateur de réaliser une tâche en réalité virtuelle aussi aisément qu'en réalité.

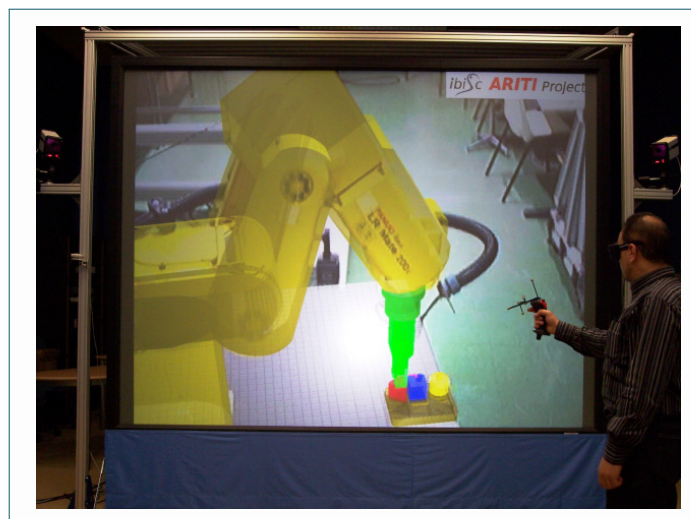


Figure 28: Application d'un guide virtuel pour guider l'utilisateur à saisir un objet avec une pince dans le cadre de la téléopération. Le guide virtuel apparaît en vert sur la photo [Otmane and Davesne, 2009].

On peut également citer les travaux de [Fitzmaurice et al](#) qui proposent ce qu'ils appellent *la navigation sûre* [[Fitzmaurice et al, 2008](#)] et que l'on peut utiliser dans le logiciel *3D's Max*. Elle consiste à assister l'utilisateur en le guidant de manière transparente lorsqu'il utilise des outils de zoom, déplacement, rotation du point de vue dans les logiciels de CAO. En effet dans ces logiciels, la maîtrise de ces outils demande beaucoup d'apprentissage et une mauvaise utilisation de ceux-ci peuvent entraîner une incompréhension de sa part lorsqu'il fait une mauvaise manipulation de ces outils.

Avec toujours la volonté d'aider l'utilisateur à se positionner ou se déplacer dans l'environnement virtuel, [[Beckhaus et al, 2002](#)] propose une assistance à l'exploration basée sur les champs de potentiels. En appliquant des valeurs plus ou moins importantes d'attraction aux objets de l'environnement virtuel, la caméra virtuelle sera plus ou moins attirée vers ceux-ci.

[[Andújar et al, 2004](#)] ont proposé, quant à eux, de calculer hors-ligne (avant la phase d'interaction 3D) un parcours, permettant d'assurer l'exploration de l'ensemble de l'environnement virtuel. Une fois le trajet déterminé, l'utilisateur, dans la phase d'interaction 3D, est contraint de suivre celui-ci.

1.4.4 Les guides haptiques

Les guides haptiques sont en quelques sortes des guides virtuels augmentés. Le terme *augmenté* désigne ici le fait que l'utilisateur ressent une force le guidant dans la réalisation de sa tâche. Ces guides peuvent être de type attractif (l'utilisateur est comme aspiré par une force) ou bien répulsif (l'utilisateur est dans ce cas-là expulsé) ou du type couloir (l'utilisateur est obligé d'avancer dans la direction du couloir, il n'a pas la possibilité de quitter ce couloir).

Ainsi, [Otmame and Davesne](#) [[Otmame and Davesne, 2009](#)] ont utilisé des guides répulsifs afin d'assister l'utilisateur lors du positionnement précis d'une pince de robot virtuel au-dessus d'un objet à saisir (Figures 30 et 29). La finalité de cette application étant d'aider l'utilisateur dans son interaction avec le robot virtuel afin de réaliser une tâche en virtuel, pour enfin, par la suite, envoyer les commandes au robot réel afin qu'il réalise la tâche en conditions réelles.

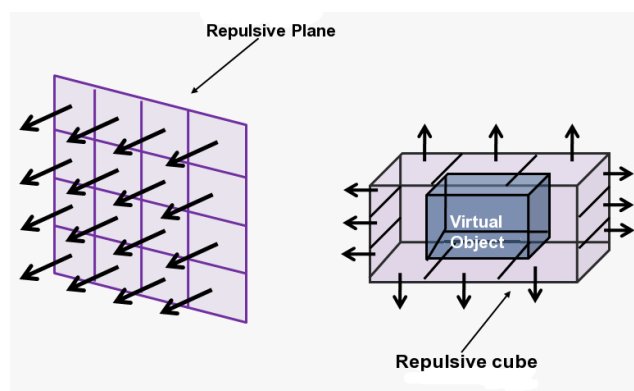


Figure 29: Représentation des guides répulsifs placés sur les différentes faces internes d'un parallélépipède [[Otmame and Davesne, 2009](#)].

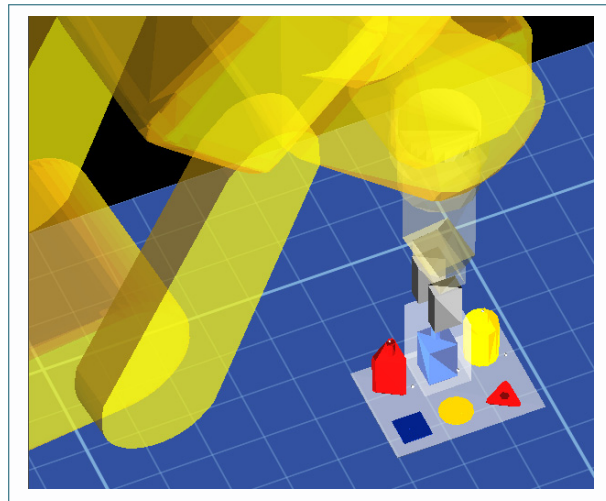


Figure 30: Application des guides répulsifs à une application de téléopération. Le guide répulsif apparaît comme un cube blanc translucide [Otmame and Davesne, 2009].

Ullah et al [Ullah et al, 2008] ont, quant à eux, proposé l'utilisation de guides haptiques attractifs et répulsifs de différentes formes (voir figure 31) afin d'assister l'utilisateur à la saisie d'objets virtuels ou bien à leur évitement.

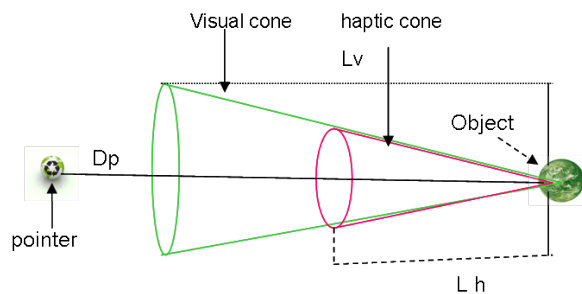


Figure 31: Schéma représentant le guide haptique proposé par Ullah et al [Ullah et al, 2008].

CONCLUSION

Toutes ces assistances ont pour but d'aider l'utilisateur à accomplir sa tâche, mais pour cela les évaluations démontrent que ces assistances ne doivent pas être intrusives et que l'utilisateur doit comprendre sur quoi elles influent pour qu'elles soient bien vécues par celui-ci.

1.5 LES ARCHITECTURES MULTI-CAPTEURS ET MULTIMODALES

1.5.1 Présentation

On parle de systèmes multi-capteurs dès qu'il y a combinaison de plusieurs capteurs, périphériques, interfaces, moyens techniques, méthodes ou stratégies pour accomplir une

tâche particulière. Dans le cas de l'interaction 3D, un système multi-capteurs se traduit par l'utilisation combinée de plusieurs interfaces sensori-motrices (exemple : tracking optique, gant de données) afin de permettre à l'utilisateur d'interagir en 3D avec un environnement virtuel.

Les systèmes multi-capteurs sont intimement liés aux systèmes multimodaux. On appelle multimodalité, le fait que l'utilisateur puisse utiliser à des instants différents ou en même temps des interfaces sensori-motrices faisant appel à ses différents sens. En effet dès lors qu'il y a multimodalité, il est nécessaire d'avoir plusieurs capteurs. C'est pour cela que dans ce chapitre, nous nous intéresserons aux architectures des systèmes multi-capteurs mais également aux architectures des systèmes multimodaux. La multiplication des capteurs est un aspect très important dans le domaine de l'interaction 3D car elle permet d'améliorer les performances des applications de RV/RA. Elle permet également de rendre l'interaction plus naturelle et intuitive car dans un système multi-capteurs on suit généralement les lois de la physique et les conventions du sens commun. La relative simplicité d'utilisation des systèmes d'interaction 3D multimodale provient de l'usage de schèmes ou de leurs variantes, c'est-à-dire des actions que l'on sait faire depuis notre petite enfance et qui ne nécessitent pas une durée d'apprentissage trop importante voire une durée d'apprentissage négligeable.

Généralement, les problèmes des systèmes d'interaction 3D multimodaux ne proviennent pas de l'utilisateur mais de l'interprétation de ses gestes. En effet, ces systèmes devant gérer plusieurs entrées pour une ou plusieurs sorties, des interprétations contradictoires peuvent apparaître dues à diverses raisons. A titre d'exemple, deux périphériques permettent de donner la position et l'orientation de la main de l'utilisateur dans l'espace, mais ils travaillent tous les deux dans des référentiels différents et avec une précision différente sur la position et l'orientation. Le système d'Interaction 3D Multimodale (I3DM) doit faire en sorte de prendre le meilleur des deux périphériques en faisant coïncider l'action réelle effectuée par l'utilisateur et l'action effectuée dans l'environnement virtuel. Le système d'I3DM doit donc choisir une unique interprétation de l'intention de l'utilisateur à partir de plusieurs interprétations différentes dues à la multimodalité. Le schéma de la figure 32 illustre cette problématique.

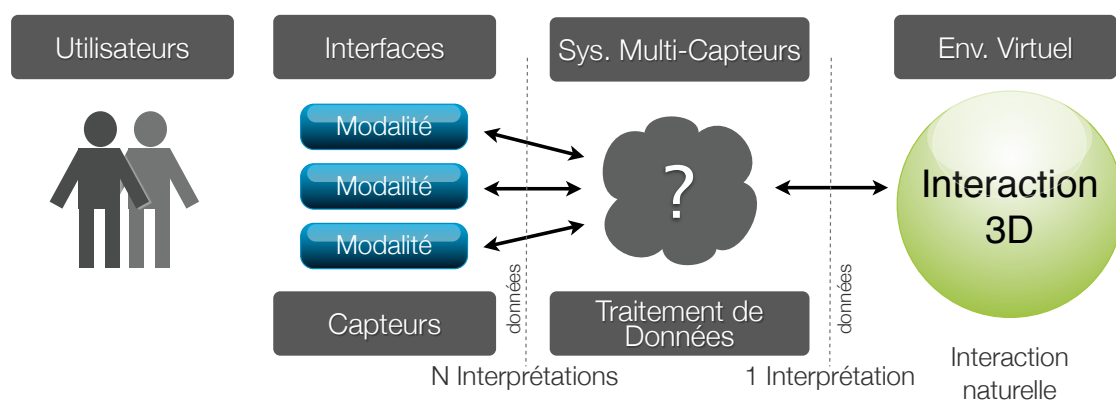


Figure 32: Problématique de l'interaction 3D multi-capteurs.

De nombreuses études ont été réalisées pour proposer des architectures de système 3D multi-capteurs, à même de gérer une multitude de périphériques et de techniques

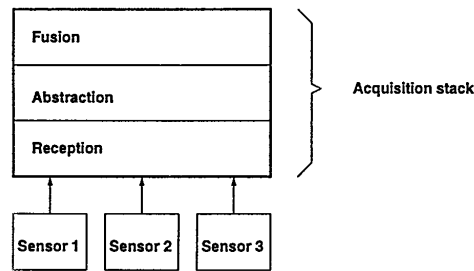


Figure 34: La pile d'acquisition conçue par Leonhardt and Magee [Leonhardt and Magee, 1998].

Intéressons-nous à ces couches, elles sont au nombre de trois :

1. *La couche de réception* : Cette couche sert à assurer la synchronisation et la distribution des capteurs de manière transparente pour l'utilisateur grâce à un bus de capteurs assurant la communication avec la pile d'acquisition.
2. *La couche d'abstraction* : Cette couche sert à unifier la représentation des données provenant des capteurs. Une telle approche permet d'ajouter de nouveaux capteurs et de les prendre en compte dès qu'ils deviennent disponibles. La représentation unifiée des données en sortie doit être indépendante des caractéristiques des capteurs.
3. *La couche de fusion* : Cette couche ne s'occupe pas d'homogénéiser les données issues des capteurs, puisque cela est réalisé par la couche d'abstraction, mais seulement à exploiter la superposition des positions provenant de plusieurs capteurs ainsi qu'à détecter les incohérences.

Un des aspects intéressant dans cette architecture est que la structure de cette pile est récursive, ce qui permet de la répéter à l'infini (voir figure 35).

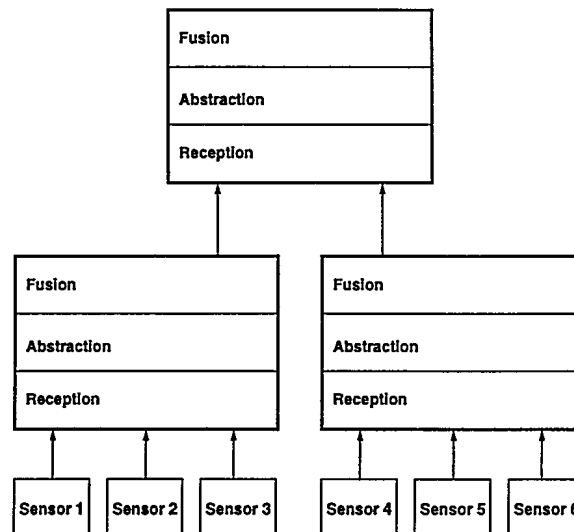


Figure 35: Un exemple de structure récursive [Leonhardt and Magee, 1998].

1.5.4 Ontologie spatiale et architecture d'interaction 3D multimodale

Dans le cadre de l'interaction 3D multimodale, j'ai retenu le travail de [Irawati et al](#) sur un modèle d'architecture d'interaction 3D multimodale basé sur l'ontologie spatiale [[Irawati et al, 2006](#)]. Toute la partie suivante fait référence à leurs travaux car ils reflètent bien les architectures de système que l'on peut rencontrer, de manière générale, dans le domaine de la réalité virtuelle.

La figure 36 présente leurs architectures d'interaction 3D multimodale. Elle est destinée à être la base pour le développement de systèmes d'interaction 3D multimodale. En utilisant cette architecture, un certain nombre de modalités d'entrées pourront être combinées afin d'interagir en 3 dimensions en utilisant la technique d'interaction 3D qui sera la plus adéquate. Nous allons détailler le rôle des différents composants et les relations qu'ils entretiennent.

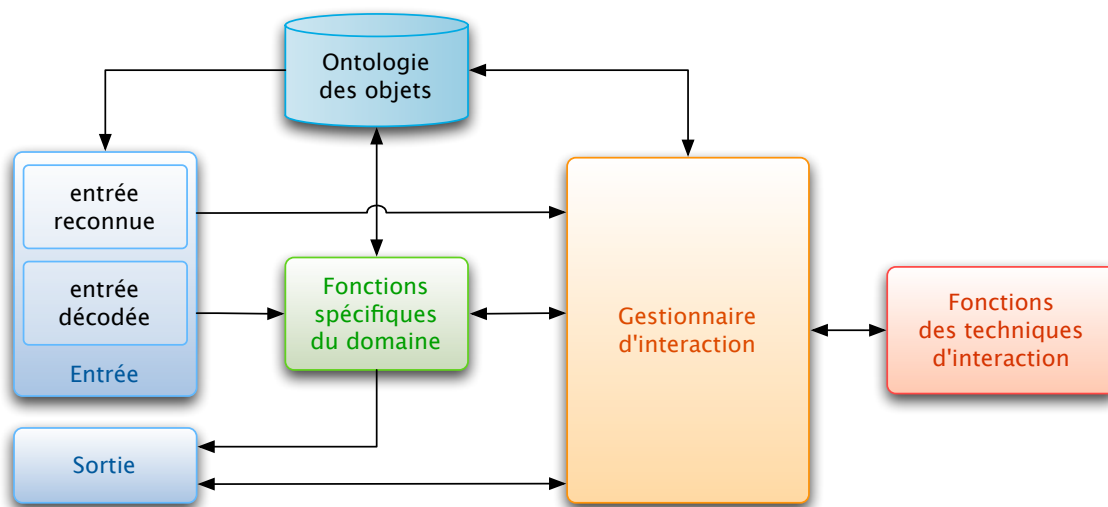


Figure 36: Architecture d'interaction 3D multimodale adaptée [[Irawati et al, 2006](#)].

1.5.4.1 L'utilisateur

L'utilisateur est celui qui agit sur les entrées et récupère les informations présentées par le système.

1.5.4.2 L'entrée

L'entrée est une entrée multimodale, pouvant recevoir divers périphériques tel qu'un clavier, une voix, une souris, un dispositif de tracking, etc. On distingue deux classes d'entrées : les entrées reconnues et les entrées décodées.

- Les entrées reconnues sont les entrées qui nécessitent un système de reconnaissance. Ce sont les entrées de type : écriture manuscrite, voix, vidéo, etc.
- Les entrées décodées sont les entrées qui nécessitent un décodage de l'information en chaînes de caractères ou de nombre de type dispositifs de tracking, boutons, signaux analogiques.

1.5.4.3 *La sortie*

La sortie est une sortie multimodale, elle aussi, pouvant être connectée à des dispositifs audio, vidéo, à retour d'effort, etc.

1.5.4.4 *Le gestionnaire d'interaction et les fonctions de techniques d'interaction*

Le gestionnaire d'interaction est le composant qui s'occupe de coordonner les données provenant des différentes entrées et d'envoyer les données nécessaires vers les sorties. Le gestionnaire d'interaction permet aussi d'utiliser les techniques d'interaction 3D pour la sélection et la manipulation (curseur 3D, Raycasting, etc.) définies dans le composant : fonctions de techniques d'interaction. Son rôle est de prendre en compte les diverses interprétations des différentes entrées afin de définir une seule interprétation pour le système. Pour l'aider dans sa tâche, on utilise l'ontologie des objets et le contexte de l'utilisateur.

1.5.4.5 *Serveur de la parole et gestionnaire de parole*

La voix est une entrée reconnue qui nécessite un moteur de reconnaissance vocal afin d'être convertie en chaîne de caractères. Le moteur de reconnaissance vocale est situé sur le serveur de la parole, afin de fonctionner il a besoin d'un fichier de grammaire contenant tous les mots pouvant être reconnus par le système. Chaque groupe de mots compose une règle. Une interprétation sémantique peut être effectuée sur la phrase reconstituée par le moteur de reconnaissance vocale afin de réaliser un traitement prenant en compte les commandes antérieures reçues en entrée. L'enregistrement effectué par le serveur de la parole est envoyé au gestionnaire de la parole qui effectuera quelques traitements supplémentaires, puis, qui enverra une interprétation au gestionnaire d'interaction. Une partie de la grammaire de la parole doit être associée avec l'ontologie des objets afin d'interpréter le sens de la commande vocale.

1.5.4.6 *Ontologie des objets*

A - ONTOLOGIE ET OBJETS VIRTUELS

Dans le cadre de l'interaction 3D, l'ontologie permet de stocker les informations à propos des objets virtuels et des relations spatiales qu'ils entretiennent. La finalité de cette utilisation est de vérifier la validité de la détection des intentions de l'utilisateur lorsqu'il interagit avec des objets virtuels. Par conséquence, l'ontologie spatiale peut être utilisée pour assister l'utilisateur quand il place et manipule des objets dans un environnement virtuel et ainsi permettre à l'utilisateur d'effectuer dans l'environnement virtuel ce qu'il ferait avec aisance dans la réalité (exemple : poser un objet sur une table). Dans cette architecture, chaque objet virtuel est défini comme un objet au sens spatial qui possède des attributs (exemple : orientation et position). On distingue plusieurs types d'objets.

- *Objet statique* : C'est tout simplement un objet qui ne peut pas être manipulé par l'utilisateur.
- *Objet interactif* : C'est un objet qui peut être manipulé par l'utilisateur. L'utilisateur peut modifier certaines propriétés de l'objet telles que sa position et son orientation.

- *Objet positionnable* : C'est un objet sur lequel l'utilisateur peut placer un autre objet (exemple : une table).
- *Objet situé horizontalement* : C'est un objet généralement positionné sur une (exemple : un verre) ou plusieurs (exemple : les quatre pieds d'une chaise) de ses surfaces horizontales.
- *Objet situé verticalement* : C'est un objet généralement positionné sur une (exemples : un cadre, un mur) ou plusieurs de ses surfaces verticales.

B - ONTOLOGIE ET COMMANDES VOCALES

L'ontologie spatiale permet entre autre de lever l'ambiguïté qui peut exister dans les commandes vocales de l'utilisateur et ainsi permet au système d'avoir une meilleure compréhension des intentions de l'utilisateur. En effet, dans le cas où l'utilisateur est dans un environnement virtuel et qu'il pointe dans la direction d'une table en disant : "*Mettez la lampe ici.*", il est possible qu'il ait sélectionné en même temps un des murs et un livre. Dans ce cas-là, il y a ambiguïté sur ce que désigne le "*ici*" car du point de vue du système, une table, un mur et un livre sont sélectionnés et par conséquent "*ici*" peut être défini par un de ces trois objets. C'est à ce stade-là qu'intervient l'ontologie spatiale. Le gestionnaire d'interaction interprète l'ontologie et ainsi comprendre que le "*ici*" ne peut être que la table dans ce cas-là, car une relation existe entre la lampe et la table mais n'existe pas entre la lampe et le mur ou même entre la lampe et le livre.

1.5.4.7 Contexte de l'utilisateur

Il s'agit tout simplement d'un historique des interactions de l'utilisateur. Cet historique est mis à jour par le gestionnaire d'interaction.

1.5.4.8 Les fonctions spécifiques du domaine

Ce composant définit des fonctions qui dépendent du domaine de l'application. Il communique avec les composants suivants : ontologie des objets, gestionnaire de la parole, gestionnaire de périphérique et gestionnaire de l'interaction.

1.5.4.9 Le gestionnaire de périphériques et serveur de périphériques

Dans le cas d'un système d'interaction 3D multimodale, nous avons vu que plusieurs interfaces sont utilisées pour retranscrire la multimodalité. Chaque interface est vue par le système comme un périphérique d'entrée (tracking optique, gant de données, etc.), de sortie (écran, HMD, enceintes, etc.) ou d'entrée/sortie (SPIDAR, bras haptique, etc.). Afin de gérer tous ces périphériques, le système a besoin d'un gestionnaire de périphériques, dont le rôle est essentiel au fonctionnement du système. En effet, il doit permettre de transmettre les différentes informations (position, orientation, boutons, etc.) provenant d'un périphérique vers le gestionnaire d'interaction, soit dans l'autre sens. Le gestionnaire de périphériques communique avec les périphériques via un serveur de périphériques qui leur est propre. Les périphériques pouvant être connectés indépendamment à différents ordinateurs, par conséquence, il est possible d'avoir plusieurs serveurs

de périphériques. La figure 37 montre les interconnexions qui existent entre le gestionnaire des interactions et les périphériques. Chaque serveur est configuré par un fichier de configuration qui définit les propriétés du périphérique.

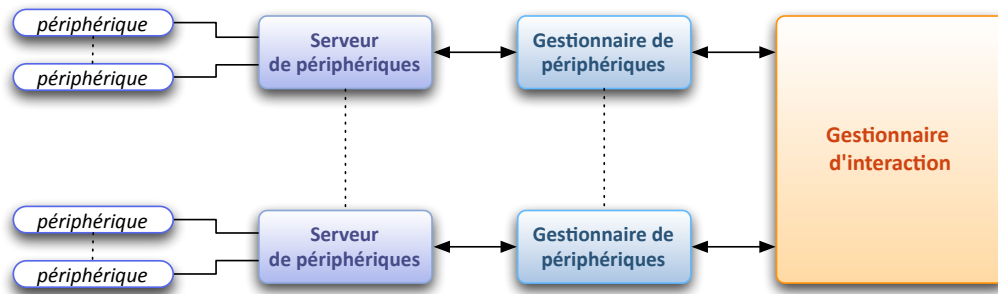


Figure 37: Relations périphériques - serveur de périphériques - gestionnaire de périphériques - gestionnaire d'interaction, adapté de [Irawati et al, 2006].

1.5.4.10 La boucle d'interaction, fonctionnement du modèle

La figure 38 représente le diagramme de la boucle d'interaction du modèle. Ce diagramme décrit le fonctionnement du modèle d'architecture d'interaction 3D multimodale proposée.

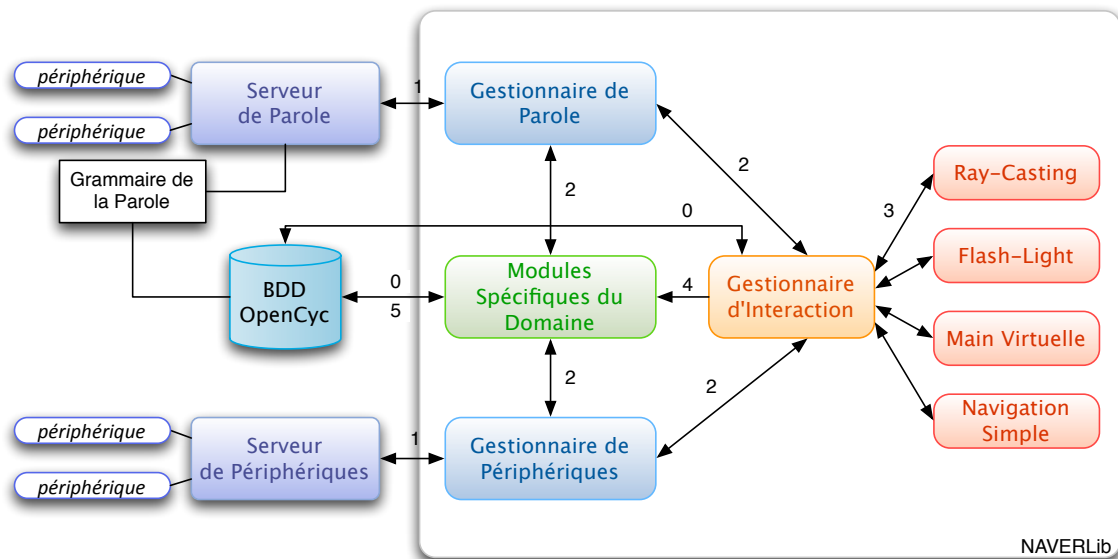


Figure 38: La boucle d'interaction adaptée de [Irawati et al, 2006].

ETAPE 0 : Cette première étape constitue l'initialisation du système. Elle consiste à établir la connexion avec la base de données à base ontologique afin de récupérer la description des objets.

Toutes les étapes suivantes font parties de la boucle d'interaction.

ETAPE 1 : D'un côté le serveur de parole effectue la reconnaissance vocale de l'entrée audio. De l'autre les serveurs de périphériques reçoivent les données des périphériques. Toutes les données reçues par les serveurs de périphériques et le

serveur de parole sont datées et envoyées respectivement vers le gestionnaire de périphériques et le gestionnaire de parole.

ETAPE 2 : Les données reçues par le gestionnaire de périphériques sont ensuite envoyées vers le gestionnaire d'interaction ou elle sont traitées comme valeurs significatives et interprétées comme pose (position/orientation/configuration) de la main, de la tête, etc. en fonction des périphériques utilisés. Ces données peuvent aussi être envoyées aux modules spécifiques du domaine. Au même moment le serveur de parole envoie son interprétation des commandes vocales de l'utilisateur à la fois au gestionnaire d'interaction et aux modules spécifiques du domaine, car les commandes vocales sont dépendantes du domaine, du contexte.

ETAPE 3 : A cette étape, le gestionnaire d'interaction envoie les données relatives aux poses au composant gérant les techniques d'interaction. Ce composant en contre partie peut renvoyer des données au gestionnaire d'interaction, par exemple la liste des objets sélectionnables si on est dans le cas d'une tâche de sélection.

ETAPE 4 : C'est à cette étape que l'interprétation des intentions de l'utilisateur est effectuée car via l'ensemble des données datées qu'il a reçu des différents composants et des fonctions spécifiques au domaine, il est à même de prendre la meilleure décision. Toutes les informations liées à l'interaction sont écrites dans un fichier de log, grâce auquel le gestionnaire d'interaction pourra mieux interpréter la prochaine commande de l'utilisateur.

ETAPE 5 : Cette dernière étape consiste à mettre à jour la base de données *OpenCyc* par les modules spécifiques du domaine.

1.5.5 Désambiguïsation mutuelle pour l'interaction 3D multimodale

1.5.5.1 Présentation

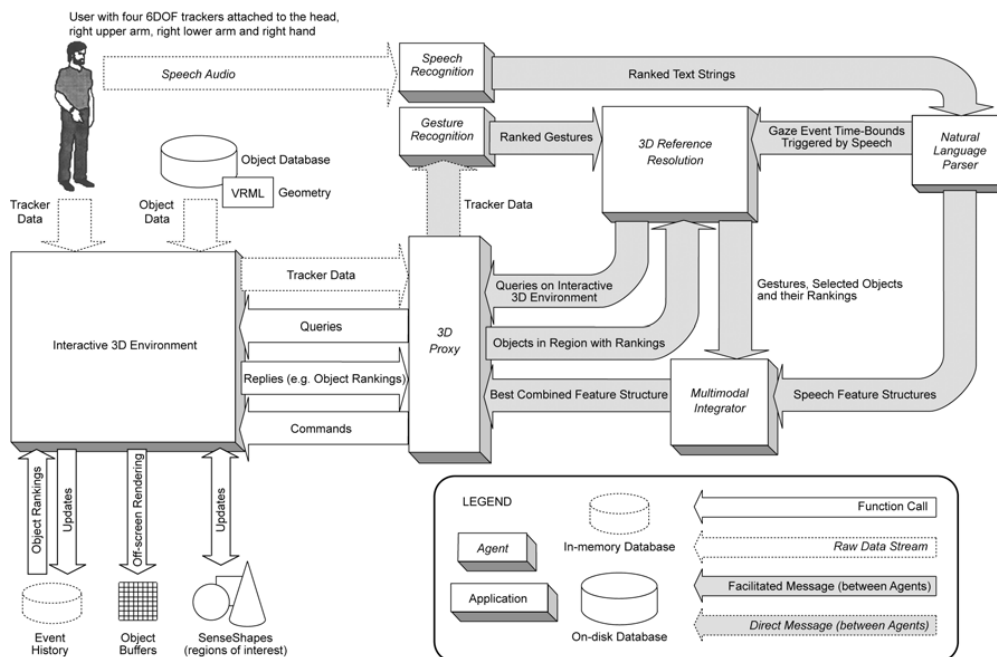


Figure 39: Architecture générale [Kaiser et al, 2003].

Les travaux de [Kaiser et al, 2003], présentés figure 39, sont une approche possible pour l'interaction 3D multimodale pour les environnements de réalité virtuelle, dans lesquelles le système doit prendre en compte la nature incertaine des sources. La partie qui m'intéresse dans ces travaux est la désambiguïsation mutuelle qui permet de lever l'ambiguïté résultante des différentes modalités et des sources d'informations. Pour cela, le système effectue une fusion de données symboliques et statistiques provenant des différentes modalités (dans leur cas : la voix, les gestes). L'architecture générale est dans l'ensemble similaire au modèle vu précédemment, néanmoins deux concepts ont suscité notre intérêt : le concept de classement et le concept d'intégrateur multimodal.

1.5.5.2 Le concept de classement

Dans cette architecture, les entrées (la voix et les gestes) passent chacune par un système de reconnaissance qui les convertit en données aisément manipulables. Ces données sont datées et classées, c'est à dire qu'on leur attribue une note représentative de leur fiabilité (exemple : la correspondance pour la voix).

1.5.5.3 Le concept d'intégrateur multimodal

L'intégrateur multimodal a pour rôle de trouver la meilleure interprétation multimodale à partir des données notées des entrées. Le principe est d'unifier les données en :

- Fusionnant les informations redondantes ou complémentaires via un ensemble de tests logiques;
- Prenant en compte l'aspect spatio-temporelles des données;
- Prenant en compte la hiérarchie des données.

1.5.6 VR Juggler

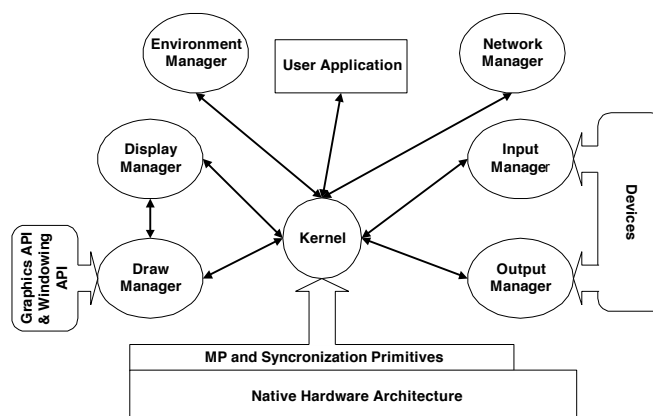


Figure 40: Architecture de VR Juggler.

VR Juggler [Just et al, 1998] est un environnement de développement d'application de réalité virtuelle. Il permet aussi bien de gérer les périphériques que le rendu visuel, l'environnement virtuel ainsi que l'aspect collaboratif.

1.5.7 VRPN: *Virtual Reality Peripheral Network*

Taylor et al ont conçu un jeu de classes utilisables, toute regroupées au sein d'une même librairie, ainsi qu'un client et un serveur permettant de communiquer avec un grand nombre de périphériques utilisés en réalité virtuelle [Taylor et al, 2001]. Cet ensemble se nomme VRPN, pour *Virtual-Reality Peripheral Network*. L'intérêt de cette approche est de créer une couche supplémentaire sur le réseau qui fait le lien entre l'application de réalité virtuelle et les périphériques. De plus, il définit une couche d'abstraction qui permet de décrire de la même manière le même type de périphérique.

1.6 CONCLUSION

Dans ce chapitre, nous avons dans un premier temps défini et présenté les multiples aspects de la réalité virtuelle. Nous avons pu voir la difficulté à donner une définition unique de la réalité virtuelle, vu que c'est un domaine vaste et pluridisciplinaire. Nous avons vu qu'en fonction de la discipline, chacune des définitions est appropriée et défendable.

Par la suite, nous nous sommes intéressés à l'interaction 3D ainsi qu'aux tâches que nous pouvons accomplir en réalité virtuelle. Nous avons pu apprécier, les problématiques de chacune de ces tâches et comment les travaux de certains chercheurs ont tenté d'y répondre par le biais de technique d'interaction 3D. Par la présentation de ces techniques, nous avons vu les métaphores et paradigmes couramment utilisées en réalité virtuelle ainsi que leur dépendance au matériel utilisé.

Enfin, nous avons porté notre attention sur les architectures multimodales en réalité virtuelle, dont le rôle est de faire le lien entre les différents périphériques et le logiciel de réalité virtuelle. On a pu voir avec ces différents travaux de recherches qu'une majorité partage des composants communs et adoptent tous un découpage en fonctions principales. Cependant, nous constatons que bien souvent pour certains aspects ils fassent appels à des composants provenant d'outils de développement tiers, ne garantissant pas une compatibilité optimale entre les différents composants, ni même une réponse adaptée à une majorité d'applications.

LES PÉRIPHÉRIQUES ET CAPTEURS D'INTERACTION UTILISÉS EN RÉALITÉ VIRTUELLE

2.1 INTRODUCTION

Un capteur est un dispositif permettant de mesurer une grandeur physique, puis de la transformer en une valeur utilisable. C'est l'élément de base utilisé dans le cadre de l'acquisition de données. On trouve différents types de fonctionnement pour les capteurs, ainsi que différentes familles de capteurs. Dans le cadre de cette recherche, nous nous limiterons aux capteurs utilisés dans le domaine de la réalité virtuelle. Les périphériques utilisent un ou plusieurs capteurs afin de permettre à l'utilisateur d'interagir avec l'environnement virtuel. Le système, lui, ne voit qu'une succession de données, provenant du périphérique, qui varient au cours du temps.

Dans le domaine de la réalité virtuelle, leur présence est essentielle pour permettre à un utilisateur d'interagir en 3D. Leur ergonomie, leur temps de réponse, leur sensibilité et leurs coûts sont des paramètres à prendre en compte lorsque l'on veut concevoir un système d'interaction 3D adapté.

De nombreuses entreprises ou équipes de recherches cherchent sans cesse à concevoir de nouveaux périphériques. Certains préfèrent cibler un besoin spécifique, mais avec le plus de précision possible. D'autres, visent plutôt à concevoir un périphérique générique qui essaie de répondre au mieux à une multitude de besoins. Les avancées technologiques et la miniaturisation permettent à ces périphériques d'être de plus en plus performants et de s'adapter de plus en plus aux différents sens de l'utilisateur. En effet, en réalité virtuelle c'est de ça dont il est question : ce type de périphérique doit être une interface entre les différents sens de l'Homme et la machine, on parle de d'interface sensori-motrice.

Ces périphériques d'interaction peuvent être classifiés en quatre catégories:

- Les périphériques de tracking,
- Les périphériques d'entrée,
- Les périphériques à retour d'effort,
- Et une dernière catégorie de plus en plus usitée, les périphériques hybrides.

2.2 LES PÉRIPHÉRIQUES DE TRACKING

Les systèmes de tracking sont des périphériques permettant de réaliser le suivi d'une ou plusieurs entités soit dans un espace 2D (une position 2D dans un plan), soit 2.5D (une position 2D et une rotation dans un plan), mais encore 3D (une position 3D dans l'espace) ou 6D (une position 3D et une orientation 3D dans l'espace). Ces entités peuvent exister sous plusieurs formes :

- Un être humain ou une partie d'un être humain,
- Une forme particulière,
- Une couleur,
- Un ou plusieurs capteurs passifs,
- Un ou plusieurs capteurs actifs,
- Ou bien une combinaison de l'ensemble des formes précédentes.

Ces systèmes sont basés sur des technologies et principes physiques différents disposant chacun d'avantages et d'inconvénients. Nous allons dans cette partie, présenter une grande partie des technologies existantes et utilisées en réalité virtuelle.

2.2.1 *Les systèmes de tracking optique*

Ces systèmes sont le plus souvent basés sur des technologies dites optiques c'est à dire des caméras ou des LASERs. Les caméras peuvent filmer des images dans la lumière visible ou bien dans le domaine des infrarouges. Nous présentons dans la suite les systèmes de tracking optique les plus usités.

2.2.1.1 *Les caméras vidéos*

Il est possible de faire du suivi d'objet à partir de caméras vidéo. Avec une seule caméra, on est capable de faire sans aucun problème du suivi 2D, voir du suivi 3D en utilisant des miroirs ou bien une variation des dimensions de l'objet. Mais pour un suivi 3D efficace et performant, permettant de récupérer les six degrés de liberté (3 translations et 3 rotations), il faut un minimum de deux caméras. Chacune des caméras voit l'objet à suivre sous un angle différent et par géométrie épipolaire entre les deux plans images on calcule la position et l'orientation de l'objet. Les propriétés de l'objet utilisées pour le suivi peuvent être sa taille, sa couleur, sa forme ou bien son déplacement. L'inconvénient d'un tel système est que le suivi est très sensible aux conditions d'éclairages ou à la présence d'autres objets similaires à l'objet que l'on veut suivre. Mais le problème le plus évident est celui de l'occultation de la caméra ou de l'objet suivi. Les avantages sont que le système est peu coûteux à mettre en place et que dans un environnement maîtrisé, le suivi peut être très précis et robuste

2.2.1.2 *Les caméras infrarouges*

De nombreux systèmes utilisent des caméras infrarouges pour le suivi d'objets. Comme pour le suivi à partir de caméras vidéo, il faut un minimum de deux caméras pour avoir un bon suivi 3D de l'objet. Un des systèmes de tracking infrarouge le plus connu est celui proposé par ART [[Advance Real-Time Tracking GmbH, 2010](#)], son fonctionnement est illustré dans la figure 41 Les avantages des caméras infrarouges par rapport aux caméras vidéos est que le tracking est beaucoup plus précis et robuste. En effet, le fait de filmer en infrarouge isole beaucoup mieux les marqueurs. Il n'y a presque pas besoin de traitements supplémentaires, pour isoler seulement l'objet à suivre. Cependant le tracking infrarouge souffre, évidemment, du même problème d'occultation. Ce problème peut être en partie compensé en rajoutant des caméras supplémentaires afin de permettre de couvrir le maximum de zones d'ombre. Mais également de la détection

d'objets non-voulus, même si pour l'infrarouge, ce cas est beaucoup plus rare. Enfin ces systèmes coûtent très chers, au fur et à mesure qu'on leur ajoute des caméras supplémentaires.

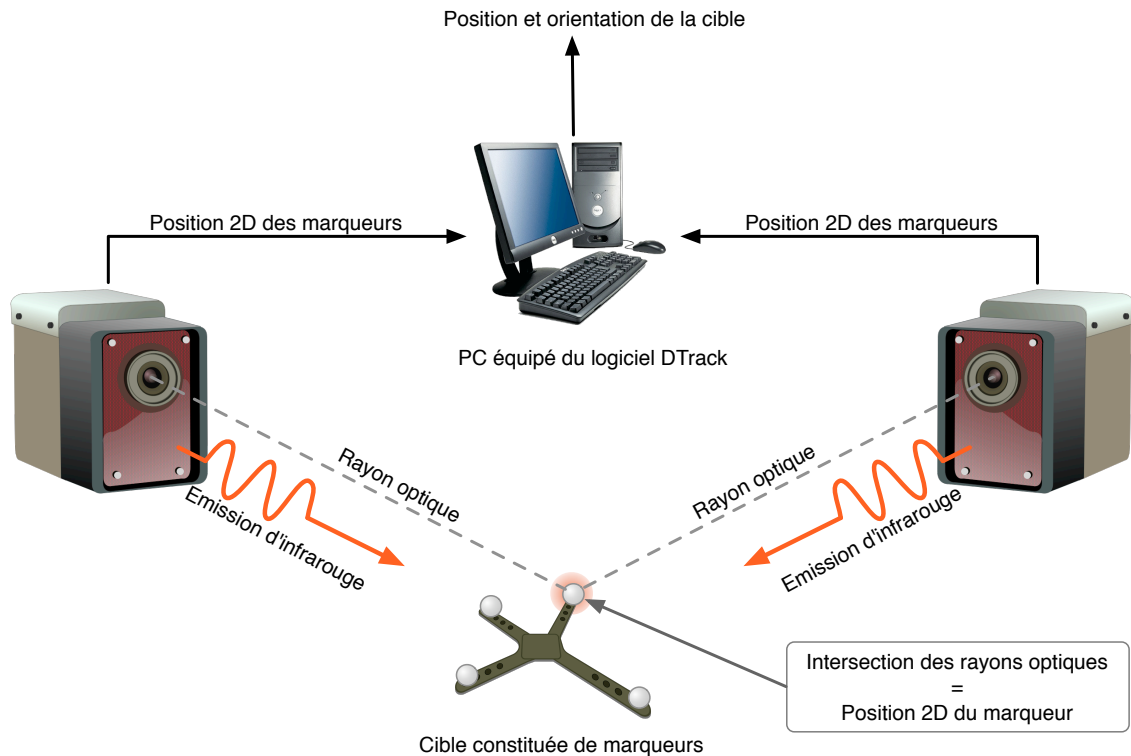


Figure 41: Le fonctionnement du système optique ART.

2.2.1.3 Les LASERS

De nombreuses recherches ont été effectuées sur l'utilisation de LASERS pour interagir avec l'environnement virtuel. Nous nous occuperons ici seulement de la partie matérielle et en aucun cas des techniques d'interaction 3D utilisées. Les pointeurs LASERS sont depuis longtemps utilisés pour désigner des éléments dans une scène. Il est très naturel de les utiliser. Ainsi, on a pensé à utiliser le LASER comme un périphérique de pointage principalement dans le domaine de l'assistance aux personnes handicapées. En effet cela permet la désignation d'objet à saisir par un bras robotisé [Chávez et al, 2008; Kemp et al, 2008]. Mais on a également pensé à utiliser le LASER pour suivre la position du spot sur un écran et ainsi interagir avec les éléments projetés sur cet écran [Kirstein and Muller, 1998; König et al, 2008; Kurz et al, 2007; Vogt et al, 2003].

Cependant un des principaux problèmes du LASER est qu'il ne projette qu'un simple spot, ce qui ne suffit pas pour déterminer une position 3D et encore moins les rotations. Pour pallier cette impossibilité, plusieurs chercheurs ont trouvé des solutions. Ainsi Latoschik et al [Latoschik et al, 2006] proposent l'utilisation d'une lentille diffractante afin de séparer le spot du laser en une vingtaine de spots, permettant ainsi de récupérer la profondeur (via l'espacement des spots) ainsi que les rotations. Dans de tels systèmes, la difficulté principale est de détecter le spot dans la scène filmée. Pour cela plusieurs méthodes ont été étudiées. La plus facile à mettre en œuvre est la détection du spot

LASER via sa couleur. L'inconvénient est qu'en suite on ne peut plus utiliser cette couleur dans la scène sous peine que le système détecte des spots là où il n'y en a pas. De plus, la moindre variation de l'éclairage de la scène peut grandement influencer le résultat de la détection. Pour palier à ce problème, une des solutions trouvées par [Chávez et al \[Chávez et al, 2008\]](#) est d'utiliser des *templates* de spots afin de les isoler au maximum de la scène filmée.

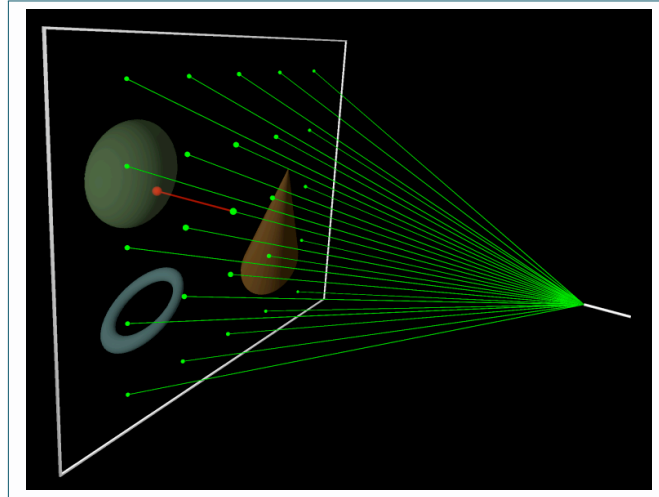


Figure 42: Le LASER utilisé avec une lentille diffractante [[Chávez et al, 2008](#)].

[Matveyev and Matveyev](#) ont utilisé un miroir afin de récupérer la profondeur, en mesurant la distance entre le spot et son reflet dans le miroir [[Matveyev and Matveyev, 2004](#)].

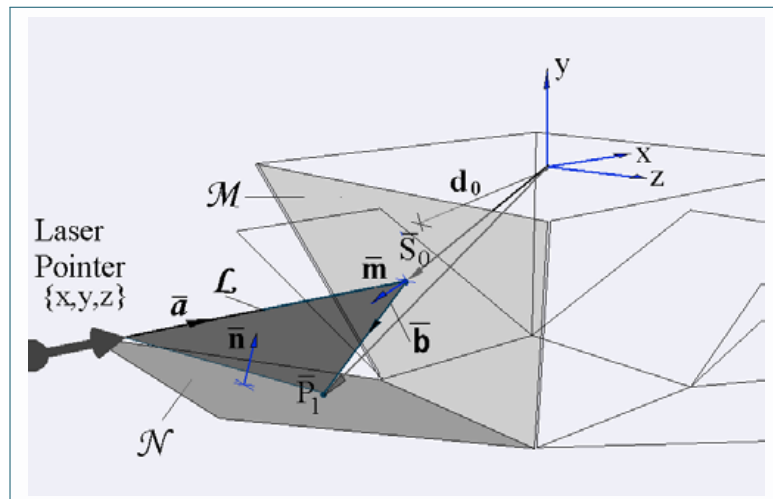


Figure 43: Schéma d'un pointeur LASER se réfléchissant sur un miroir [[Matveyev and Matveyev, 2004](#)].

2.2.2 Le tracking magnétique

Il existe également un système de tracking basé sur les variations du champ magnétique. Ces systèmes sont basés sur au moins un ensemble de 3 bobines formant un

repère 3D. Ces systèmes coûtent beaucoup moins chers que ceux basés sur l'infrarouge, mais leur précision est beaucoup moins importante. Généralement de tels systèmes existent sur des petits volumes. Ce problème de précision provient du problème de distorsions, dont souffre ce type de système de tracking, sur l'estimation de la position et de l'orientation dues aux perturbations du champ magnétique. Ce phénomène est accentué en présence de matériaux ferreux ou bien d'appareils électroniques émettant de forts champs électromagnétiques. Ce type de système de tracking est donc très sensible à l'environnement dans lequel il est situé.

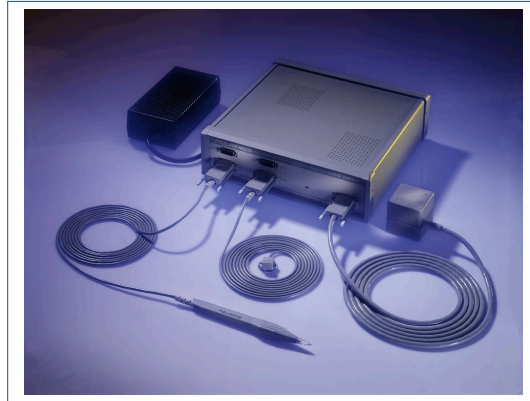


Figure 44: Un exemple de dispositif de tracking magnétique, le *Fastrack* de Polhemus.

2.2.3 Les centrales inertielles et les accéléromètres

Il est également possible de faire du suivi d'objet à partir de capteurs mesurant l'accélération instantanée sur x , y et z et la gravité. Ces capteurs sont appelés des accéléromètres. Lorsqu'un tel capteur est placé sur un objet, on est à même de mesurer les accélérations instantanées de celui-ci. À partir de la mesure des accélérations, on est capable par double intégration de déduire son orientation et sa position relative dans l'espace. Cela va de soi que l'estimation des orientations est plus précise dans le cas où la position de l'accéléromètre est quasi-immobile, il fonctionne ainsi comme un inclinomètre. L'estimation de la position relative, quant à elle, dérive très rapidement dût à l'imprécision du capteur ainsi qu'à la non connaissance des constantes d'intégrations.

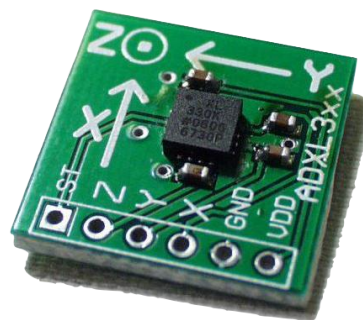


Figure 45: Un exemple d'accéléromètre.

2.2.4 Le tracking 3D à ultrasons

Ce système est la combinaison d'un système de tracking magnétique et d'un scanner ultrasonique (le même dispositif que ceux utilisés pour réaliser des échographies). Le tracking magnétique permet dans ce cas-là de localiser dans un espace 3D les images acquises par le scanner ultrasonique et ainsi de reconstruire une image 3D.

2.3 LES GANTS DE DONNÉES

Une multitude de périphériques d'interaction 3D sont basés sur des gants, la main étant la partie du corps humain, que nous utilisons le plus pour interagir avec ce qui nous entoure. Les gants de données sont composés de capteurs de flexions ou de l'écartement des doigts. L'ensemble des flexions relevées nous informe sur la configuration de la main de l'utilisateur.

Certains gants tels que les 5DT Data Glove Ultra de [Fifth Dimension Technologies](#) (représenté figure 46) sont équipés de capteurs optiques mesurant la quantité de lumière en sortie de fibres optiques fixées sur chacun des doigts. Plus la fibre optique est courbée et plus la lumière est atténuée en sortie. La quantité de lumière observée renseigne sur la flexion de chacun des doigts. D'autres gants, tels que le CyberGlove de [Cyber Gloves Systems](#), qui est représenté sur la figure 47, utilisent plutôt des matériaux sensibles à la flexion permettant de déterminer ainsi la configuration de la main.



Figure 46: Le Data Glove Ultra de [Fifth Dimension Technologies](#).



Figure 47: Le gant CyberGlove de [Cyber Gloves Systems](#).

D'autres sont utilisés pour véhiculer un retour d'information vers l'utilisateur, tel qu'une force ou un retour vibro-tactile. Ainsi, le CyberGrasp de [Cyber Gloves Systems](#)

avec son exosquelette placé sur les doigts de l'utilisateur permettent de ressentir la taille ou les contours d'un objet virtuel. Pour simuler les sensations du touché de certaines matières, cette même entreprise a développé le gant CyberTouch de [Cyber Gloves Systems](#). Ce gant utilise des vibreurs qui lorsque qu'ils fonctionnent dans une certaine plage de fréquences peuvent restituer les micro-vibrations provoquant la sensation de toucher certaines matières, telles que de l'eau ou du bois, ou de l'acier.

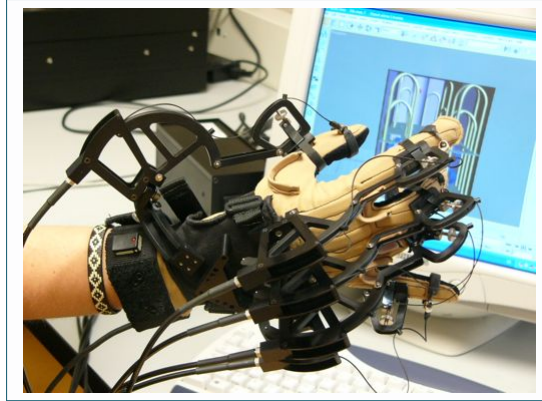


Figure 48: Le gant CyberGrasp de [Cyber Gloves Systems](#).



Figure 49: Le gant CyberTouch de [Cyber Gloves Systems](#).



Figure 50: L'accele glove [[Hernandez-Rebollar and Kyriakopoulos, 2002](#)].

L'accele glove [[Hernandez-Rebollar and Kyriakopoulos, 2002](#)] est un gant sur lequel ont été fixés plusieurs accéléromètres. Il y en a un sur chaque doigt ainsi qu'un dernier à hauteur du poignet. Il a été conçu pour trois applications : le contrôle d'une main virtuelle, les icônes d'un bureau ainsi qu'un clavier virtuel en utilisant les vingt-six gestes de l'alphabet du langage des signes américain.

Le principal avantage de tous ces périphériques est qu'il est généralement aisé pour l'utilisateur de les utiliser dans la mesure où il interagit avec sa main en réalité virtuelle comme il le ferait dans la vie réelle. Cependant, les matériaux utilisés, la taille ou le

poids des gants, ainsi que la gêne occasionnée par les câblages et éventuelles batteries font que ces périphériques peuvent très rapidement être jugés comme anti-ergonomiques ou même désagréables à l'utilisation.

2.4 LES PÉRIPHÉRIQUES À RETOUR D'EFFORT

2.4.1 *Les bras haptiques*

Les bras haptiques sont des dispositifs haptiques qui prennent la forme d'un bras mécanique sur lesquels sont fixés des moteurs ou des vérins aux articulations afin de transmettre des forces sur l'effecteur situé au bout du bras. Ils permettent également de récupérer une position 3D, voir une orientation qui sont relativement précises. Parmi les modèles de bras haptiques, on trouve le PHANToM de [Sensable Technologies](#) (figure 51) ou bien le Virtuose de [Haption](#) (figure 52).



Figure 51: Un bras haptique PHANToM de [Sensable Technologies](#).



Figure 52: Le Virtuose de [Haption](#).

Les avantages que présente ce type de périphériques sont la sensation de tenir un outil qu'ils procurent, ainsi que leur retour d'effort. En revanche, leur espace de travail est souvent réduit. De plus, ils sont assez imposants et visibles ce qui généralement diminuent l'effet immersif lorsqu'ils sont utilisés devant un ou plusieurs écrans. On peut également noter leur manque de sensibilité au niveau du retour d'effort.

2.4.2 Le SPIDAR

Le SPIDAR [Kim et al, 2000, 2002] est à la fois un périphérique de tracking, il permet de connaître la position et l'orientation de la main de l'utilisateur dans l'espace mais aussi un périphérique à retour haptique. Il est composé d'une structure cubique sur laquelle des couples moteur/encodeur sont placés à chacune de ses extrémités. A chacun des moteurs est attachée une ficelle, les huit ficelles convergent vers un effecteur qui peut être une croix, un anneau, ou une sphère. Au démarrage de ce périphérique, chacun des moteurs enroulent leur ficelle respective afin d'exercer une tension minimale sur chacune d'elles. La somme vectorielle de ces tensions nous fournit la tension exercée sur l'effecteur. Lorsque l'on bouge l'effecteur, la quantité de ficelle enroulée perçue par les différents encodeurs permet d'établir un système de 8 équations à 6 inconnues qui lorsqu'il est résolu nous informe sur la position et l'orientation de l'effecteur. Pour exercer une force sur l'effecteur, il suffit de faire varier le couple de chacun des moteurs et ainsi de modifier la tension sur chacune des ficelles et par conséquent la tension résultante sur l'effecteur, provoquant ainsi un déplacement de celui-ci.

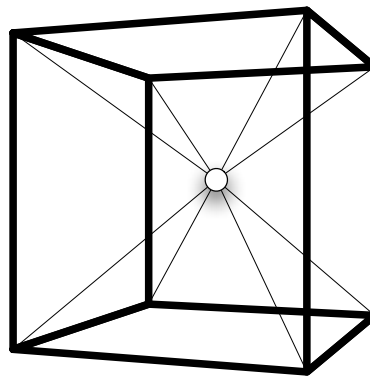


Figure 53: Structure d'un SPIDAR.

Un des principaux avantages du SPIDAR est le volume important couvert par l'espace de travail. Mais la sensation d'immersion est aussi meilleure que celle fournie par les bras haptiques, car une fois devant un écran, l'utilisateur ne distingue plus les ficelles, ce qui permet d'ignorer totalement la présence du SPIDAR.

2.5 LES PÉRIPHÉRIQUES HYBRIDES

La Wiimote est une manette de console de jeux-vidéo, développée par Nintendo® [Nintendo®, 2010], possédant diverses fonctionnalités. Elle est composée de deux accéléromètres 3 axes, divers boutons, un vibreur, un haut-parleur et d'une connectivité bluetooth. Elle est équipée d'une caméra infrarouge qui lui permet de se repérer dans l'espace lorsqu'elle est pointée en direction d'une source émettant des infrarouges. Il existe une multitude de bibliothèques de programmation afin de faire communiquer la Wiimote avec un ordinateur. Ceci a permis d'étendre l'utilisation de la Wiimote au domaine de la réalité virtuelle, offrant ainsi un périphérique hybride peu coûteux mais néanmoins très efficace.



Figure 54: La Wiimote de Nintendo® en main.

Le Bluewand est un périphérique hybride conçu par Fuhrmann et al [Fuhrmann et al, 2003] est de constitution assez similaire à la Wiimote.

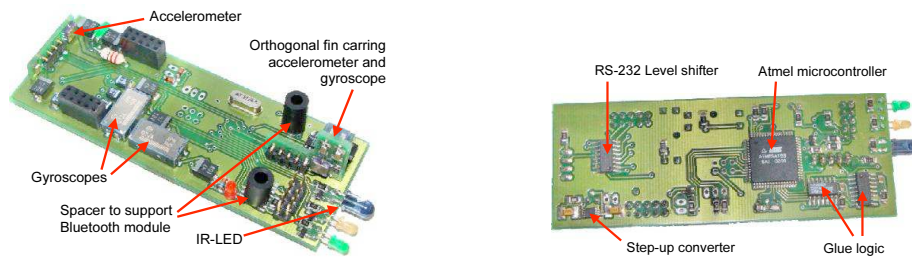


Figure 55: Les composants du BlueWand [Fuhrmann et al, 2003].

2.6 CONCLUSION

Nous venons de voir avec ce chapitre la multitude de périphériques utilisés en réalité virtuelle. Tous ces dispositifs sont basés sur des technologies différentes ayant chacune leurs avantages et inconvénients. Cela nous a permis de pointer du doigt les principaux défauts de ces périphériques ainsi que les problèmes qui peuvent apparaître lorsque l'on veut en utiliser plusieurs simultanément ou bien réaliser des combinaisons de périphériques. Une autre constatation que nous avons faite est que la tendance actuelle est de concevoir des périphériques hybrides permettant à la fois de faire du suivi 3D mais aussi de placer d'autres capteurs dessus pour déterminer d'autres intentions. Cette même tendance se retrouve dans l'industrie des jeux-vidéo qui commence à lorgner du côté de l'immersion du joueur dans l'environnement virtuel. Cependant, on peut penser que la multiplication de tous ces périphériques peut nuire à l'immersion de l'utilisateur, en particulier lorsque le périphérique ne ressemble pas à un objet ou un outil que l'utilisateur a l'habitude d'utiliser.

Mais le problème majeur est la calibration de ces périphériques. En effet, afin de permettre une utilisation efficace et performante de ceux-ci, il est primordial de connaître leurs caractéristiques, avantages et inconvénients. Tout cela dans le but de proposer,

dans la mesure du possible, un moyen de maximiser la confiance accordée dans ces périphériques ainsi que leur efficacité. C'est ce que nous présentons dans le chapitre 3.

LA CALIBRATION DES SYSTÈMES DE TRACKING 3D EN RÉALITÉ VIRTUELLE

3.1 INTRODUCTION

Les systèmes de tracking 3D en réalité virtuelle sont des dispositifs comprenant généralement un ensemble de capteurs. L'ensemble de ses capteurs délivre une information qui doit être traitée par le système afin de calculer une position 3D et éventuellement une orientation. Comme tout système multi-capteurs, ce système nécessite donc une phase de calibration pour qu'il puisse fonctionner de manière optimale. La calibration est une étape très importante pour un système de tracking. Cette étape se doit d'être exécutée avec la plus grande attention car son issue a une grande influence sur les performances et la précision des mesures effectuées via le système.

Nous nous limitons dans ce chapitre aux systèmes de tracking principalement 3D (seulement la position). Cela ne nous empêche pas d'explorer les méthodes de calibration pour d'autres types de tracking (2D et 6D) afin de souligner certains de leurs aspects qui sont susceptibles de nous intéresser. Cette étude de l'existant ne nous limite pas seulement au domaine de la réalité virtuelle, mais nous amène souvent aussi à étudier des travaux dans d'autre domaine de recherche, comme par exemple, le domaine de l'imagerie médicale car la médecine utilise souvent des bras robotisés et/ou des systèmes de tracking à ultrasons.

3.2 LA CALIBRATION D'UN SYSTÈME DE TRACKING

Les systèmes de tracking sont devenus de plus en plus importants depuis qu'ils sont majoritairement utilisés comme périphériques d'entrée dans les systèmes d'interaction 3D. Le rôle d'un système de tracking est de retourner la position 3D d'une entité dans l'espace : soit seulement la position (espace 3D), soit la position et l'orientation (espace 6D).

Dans le cas d'un système de tracking idéal, la position de l'entité retournée par le système est égale à la position réelle de l'entité dans l'espace de coordonnées adéquat. Mais dans la réalité, il n'existe pas de système de tracking parfait et par conséquent, un décalage peut être présent entre la localisation perçue par le système et la localisation réelle. Ce décalage existe sous la forme de distorsions. À cela, il faut rajouter que l'espace de travail du système est très souvent plus petit que celui permis en théorie par le système. Ces distorsions sont souvent une fonction dépendante de la distance à laquelle l'entité à suivre est située des sources de tracking. Ces distorsions peuvent souvent être corrigées/-compensées. On appelle cette opération : la calibration.

3.3 LES PROTOCOLES EXPÉRIMENTAUX DE CALIBRATION DES SYSTÈMES DE TRACKING

Pour les systèmes de tracking renvoyant une position, une grille de calibration ou une mire est souvent utilisée afin de déterminer les caractéristiques intrinsèques du système et qu'à partir de celles-là, le système corrige la position. Une grille de calibration, consiste à établir une liste de points dont la position ou la disposition est connue, en vue de confronter les positions réelles relevées par le capteur avec les positions théoriques (celles de la grille de calibration). La confrontation nous permet de caractériser ainsi le capteur et d'en extraire ses problèmes, ses erreurs, sa précision, etc.

Autant pour les systèmes de tracking 2D, comme par exemple, le suivi vidéo via une caméra, ou bien les dispositifs infrarouges, ce procédé est depuis bien longtemps maîtrisé. Il en est tout autre pour les capteurs 3D. En effet il est très difficile de mettre en œuvre une grille de calibration en 3D car on ne peut généralement pas contraindre un objet sur plusieurs positions 3D précises. Plusieurs travaux de recherche ont été menés afin de réaliser des protocoles de calibration de systèmes de tracking 3D.

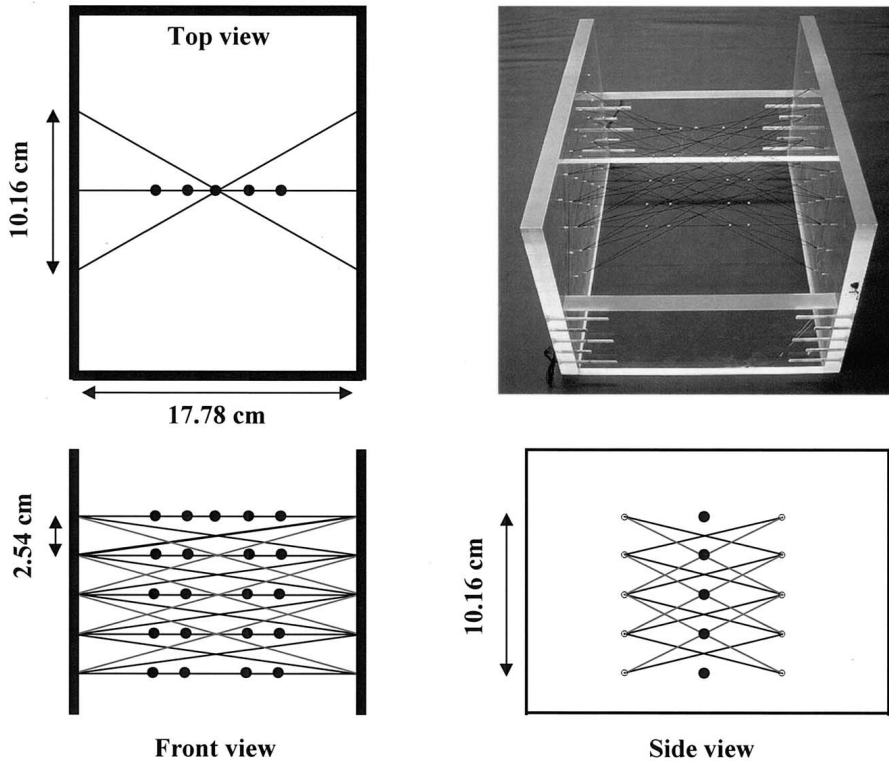


Figure 56: Disposition du plan de calibration de Leotta [Leotta, 2004].

En effet, Leotta [Leotta, 2004] a proposé d'utiliser des ficelles tendues sur lesquelles sont disposés des anneaux afin de servir de guide haptique et ainsi contraindre des mouvements en 3D. Son utilisation a été prévue pour calibrer un système de tracking à ultrason afin de guider la main de l'utilisateur à scanner la tête d'un patient et ainsi améliorer la précision et l'exactitude de l'imagerie 3D résultante. Une représentation de son plan de calibration est représentée sur la figure 56.

Une autre voie explorée est celle de l'utilisation d'un bras robotisé, voire même de robot afin d'utiliser généralement leur grande précision et/ou répétabilité pour comparer les positions relevées avec le capteur à calibrer et les positions auxquelles elles devraient être situées.

Une méthode couramment utilisée consiste à prendre un objet étalon, c'est-à-dire un objet dont les dimensions ou propriétés sont connues, établir des mesures de cet objet étalon avec le système de tracking à évaluer et à comparer les valeurs recueillies par rapport à celles connues. Ainsi [Barratt et al](#) ont utilisé un objet étalon (voir figure 57), sur lequel des petits marqueurs en métal ont été placés précisément, afin d'évaluer la précision du capteur à ultrasons [[Barratt et al, 2001](#)].

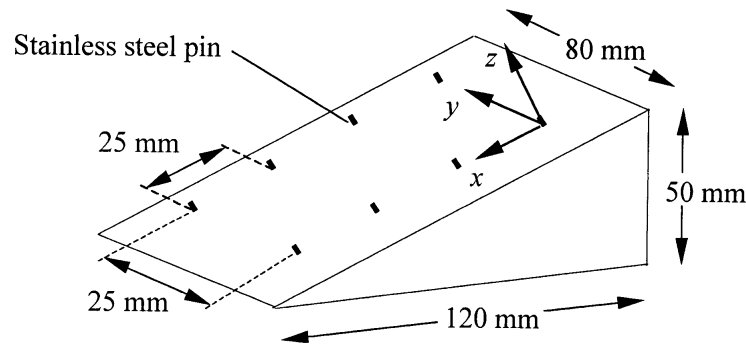


Figure 57: L'objet étalon utilisé par [Barratt et al](#) [[Barratt et al, 2001](#)].

3.4 LES MÉTHODES DE RÉGRESSION POUR LA CALIBRATION

De nombreuses approches permettent d'effectuer une calibration, elles sont généralement issue du domaine de la régression. La régression est une approche statistique, visant à analyser la relation d'une variable par rapport à une ou plusieurs autres. Cette approche consiste à proposer un modèle mathématique permettant d'estimer au mieux les données en sortie par rapport aux données observées à partir des données en entrées. On trouve des méthodes de régression linéaire et non-linéaire.

3.4.1 Les méthodes de régression linéaire

Parmi les méthodes de régression linéaire, la méthode des moindres carrés est la plus connue. Elle permet d'effectuer la comparaison entre des données relevées et une fonction mathématique. Les données relevées possèdent des erreurs qu'il est souvent nécessaire d'étudier. L'ajustement par les moindres carrés permet de déterminer la fonction mathématique $f(x_i, \theta)$ qui s'approche le plus des données observées en entrée : x_i et les données observées en sortie y_i . $f(x_i, \theta)$ représente donc les données estimées par le modèle.

Le principe est le suivant : on cherche à déterminer les paramètres θ optimaux qui minimise la quantité $S(\theta)$, c'est-à-dire la somme quadratique des déviations des mesures aux prédictions :

$$S(\theta) = \sum_{i=1}^N (y_i - f(x_i, \theta))^2 = \sum_{i=1}^N r_i^2(\theta) \quad (3.1)$$

Avec :

$S(\theta)$, la quantité à minimiser, représentative de la distance entre les données observées et les données estimées par le modèle.

$r_i(\theta)$, représentant ce que l'on appelle les résidus au modèle, c'est-à-dire l'écart entre les données observées et les données estimée par le modèle.

Si l'on connaît l'écart-type du bruit associé à chaque mesure y_i , on utilise la quantité ci-dessous, dite du khi-deux, qui prend en compte une pondération des mesures. Une mesure sera d'autant plus pondérée que son incertitude sera faible.

$$\chi^2(\theta) = \sum_{i=1}^N \left(\frac{y_i - f(x_i; \theta)}{\sigma_i} \right)^2 = \sum_{i=1}^N w_i (y_i - f(x_i; \theta))^2 \quad (3.2)$$

Avec :

$w_i = \frac{1}{\sigma_i^2}$, le poids de la mesure i .

3.4.2 Les méthodes de régression non-linéaire

3.4.2.1 Les réseaux de neurones pour la régression non-linéaire

A - PRÉSENTATION GÉNÉRALE DES RÉSEAUX DE NEURONES

Un réseau de neurones formels est un modèle, s'inspirant des vrais neurones biologiques, ceux du cerveau humain. Les réseaux de neurones, ou réseaux neuronaux, appartiennent à la fois au domaine de l'intelligence artificielle ainsi qu'à celui des applications statistiques. Ils permettent de concevoir des systèmes capables d'apprendre. Cet apprentissage est régi par le principe de l'induction, c'est-à-dire l'apprentissage par expérience.

B - HISTORIQUE DES RÉSEAUX DE NEURONES

Le premier modèle de neurone formel a été proposé par McCulloch and Pitts [McCulloch and Pitts, 1943] dans le but de modéliser le cerveau humain. La première méthode d'apprentissage : la règle de Hebb a, quant à elle, été publiée quatre ans plus tard. Les premiers réseaux de neurones formels font leurs apparitions à la fin des années 50, début des années 60 avec le perceptron [Rosenblatt, 1958] ou adaline [Widrow and Angell, 1962]. A cette époque ils sont utilisés pour réaliser des portes booléennes. L'engouement pour les réseaux de neurones disparaît lorsque Minsky conclut dans un rapport à l'inefficacité de ceux-ci pour les systèmes non-linéaires. Les réseaux de neurones redécouvrent un essor à partir des années 80, grâce à la méthode de rétro-propagation du gradient ainsi que l'incroyable augmentation de la puissance de calcul.

C - DOMAINES D'APPLICATION DES RÉSEAUX DE NEURONES

Les applications des réseaux de neurones sont très nombreuses et couvrent de nombreuses applications (reconnaissance de formes, prédiction, identification, régression, filtrage, commande, classification) qui couvrent de multiples domaines (finance, intelligence artificielle, aéronautique, robotique, etc.).

D - MODÈLE MATHÉMATIQUE DU RÉSEAU DE NEURONES

Nous présentons dans cette partie, les différents modèles d'un réseau de neurones, en commençant du simple neurone jusqu'au réseau d'un ensemble de neurones

Neurone simple avec entrée scalaire Un neurone simple peut être modélisé soit avec une seule entrée scalaire x (figure 58), soit avec une entrée scalaire x à laquelle on additionne un biais b (figure 59).

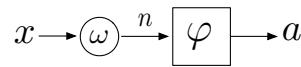


Figure 58: Un neurone simple sans biais.

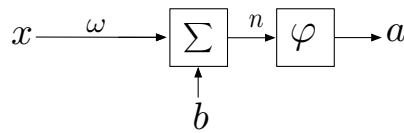


Figure 59: Un neurone simple avec biais.

Dans ces figures, une flèche représente une connexion et un carré représente une fonction. L'entrée scalaire x est multipliée par un poids ω ce qui nous donne une entrée pondérée ωx .

- Dans le cas du neurone simple sans biais, cette entrée pondérée est la variable de la fonction de transfert φ , parfois appelée fonction d'activation. En sortie de la fonction de transfert on obtient :

$$a = \varphi(\omega x) \quad (3.3)$$

- Dans le cas du neurone simple avec biais, on additionne le biais b à l'entrée pondérée ωx . On obtient donc en sortie de la fonction de transfert :

$$a = \varphi(\omega x + b) \quad (3.4)$$

Les fonctions de transfert peuvent être de différentes sortes. Les plus utilisées sont les fonctions échelon, sigmoïde et linéaire.

Il faut savoir que le principe général d'un réseau de neurones est basé sur l'ajustement du biais b et du poids ω de chacun des neurones de manière à agir sur leur sortie et on obtient donc en sortie du réseau de neurones le comportement souhaité.

Neurone simple avec une entrée vectorielle Dans le cas d'un neurone avec une entrée vectorielle X ($X = [x_1, x_2, \dots, x_M]^T$) de dimension M , à chaque entrée x_i est associée un poids ω_i définissant un vecteur de poids W ($W = [\omega_1, \omega_2, \dots, \omega_M]^T$) de dimension M .

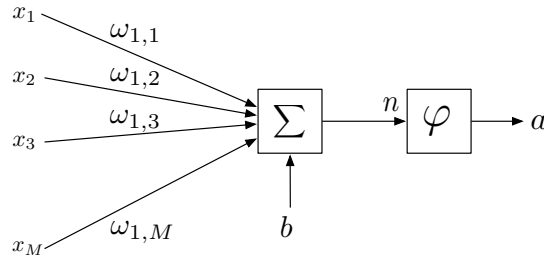


Figure 60: Un neurone avec une entrée vectorielle.

On obtient donc, avant la fonction de transfert, une entrée n de la forme :

$$n = \omega_{11}x_1 + \omega_{12}x_2 + \dots + \omega_{1M}x_M + b \quad (3.5)$$

En écriture matricielle :

$$n = WX + b \quad (3.6)$$

En sortie de la fonction de transfert φ on récupère donc :

$$a = \varphi(n) = \varphi(WX + b) \quad (3.7)$$

Réseau de neurones La figure 61 représente un réseau de neurones à 3 couches à N^0 entrées et N^3 sorties.

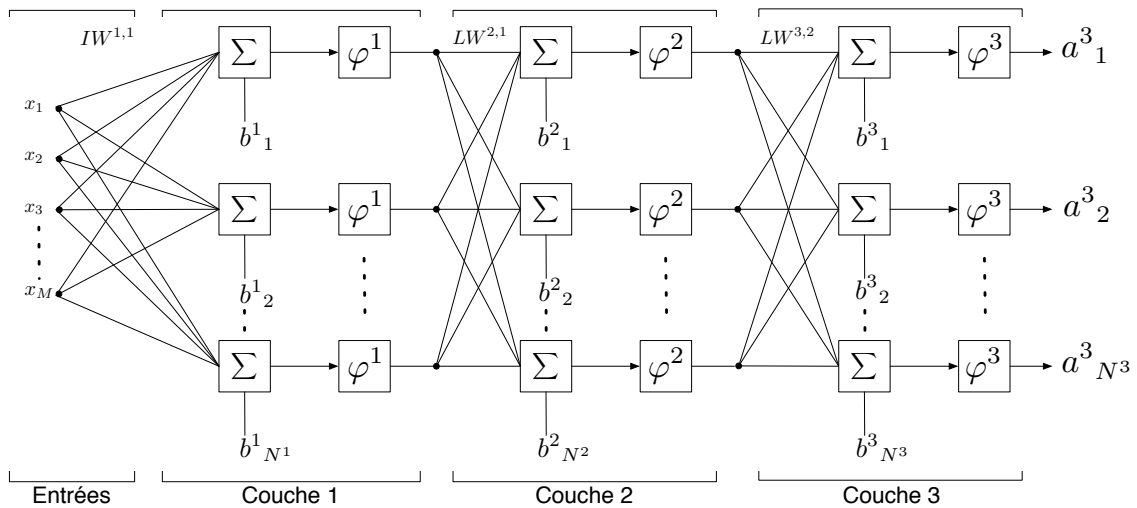


Figure 61: Un réseau de neurones à 3 couches.

Chaque sortie de couche : a^1 , a^2 et a^3 ont pour équation:

$$\begin{cases} a^1 = \varphi^1(IW^{1,1}X + b^1) \\ a^2 = \varphi^2(LW^{2,1}a^1 + b^2) \\ a^3 = \varphi^3(LW^{3,2}a^2 + b^3) \end{cases} \quad (3.8)$$

Et donc en remplaçant les a^1 et a^2 par leurs expressions respectives dans l'expression de a^3 , la sortie finale du réseaux de neurones peut s'écrire:

$$a^3 = \varphi^3(LW^{3,2}(\varphi^2(LW^{2,1}(\varphi^1(IW^{1,1}X + b^1)) + b^2)) + b^3) \quad (3.9)$$

Phase d'Apprentissage L'apprentissage permet dans le cadre des réseaux de neurones de déterminer les valeurs des paramètres suivants : les biais et les poids pour chacun des neurones. On distingue les réseaux de neurones à apprentissage supervisé et non supervisé. Lorsqu'il est à apprentissage supervisé, le réseau de neurone doit converger vers un état final précis. Lorsqu'il est non supervisé, le réseau de neurone est libre de converger vers un état stable.

L'apprentissage peut être réalisé en ligne ou hors ligne. Par le terme *en ligne*, on entend que le réseau de neurones met à jour les valeurs des différents paramètres au fur et à mesure de la simulation et ainsi parfaire son apprentissage. L'inconvénient d'un tel apprentissage est qu'il nécessite un temps de calcul et des ressources assez importantes. Lors d'un apprentissage hors ligne, on applique un algorithme qui s'exécute tant que l'ensemble des biais et des poids n'aura pas été déterminé afin que le réseau de neurones se comporte de la manière souhaitée. Une fois l'apprentissage terminé on simule le réseau de neurones et on fige les paramètres de celui-ci. Les avantages de cette méthode d'apprentissage est la rapidité de simulation du réseau de neurones.

Utilisation Une utilisation typique d'un réseau de neurone se réalise en quatre étapes:

1. Rassemblement des données pour l'apprentissage.
2. Création et configuration du réseau de neurones.
3. Apprentissage du réseau de neurones.
4. Simuler la réponse du réseau de neurones aux nouvelles entrées, c'est à dire l'étape de généralisation.

On répète autant de fois que nécessaire les étapes 2 à 4, tant que l'on n'a pas déterminé la meilleure configuration (type de réseau, nombre de neurones, nombre de couches, ...) du réseau de neurones. Généralement ces paramètres sont fixés de manière empirique, par ailleurs, c'est un des problèmes de l'utilisation des réseaux de neurones.

3.4.2.2 Réseau de neurones à rétro-propagation

Introduction La rétro-propagation, ou backpropagation en anglais, est une généralisation de la règle d'apprentissage de Widrow-Hoff aux réseaux multicouches. Le principe est le suivant : on utilise des vecteurs d'entrée et leurs vecteurs de sortie désirés afin de réaliser l'apprentissage du réseau. Cet apprentissage est effectué tant que:

- L'approximation de la fonction n'a pas été déterminée,
- La correspondance entre les vecteurs d'entrée et de sortie n'a pas été réalisée,

N'importe quel réseau modélisé à partir de biais, d'une couche avec une fonction de transfert de type sigmoïde et d'une couche de sortie avec une fonction de transfert linéaire, peut être à même d'approximer n'importe quelle fonction mathématique possédant un nombre fini de discontinuités. C'est ce qu'on appelle un approximateur universel.

La rétro-propagation est un algorithme de type descente de gradient. On parle de rétro-propagation car cela fait référence à la manière dont le gradient est calculé. Il existe beaucoup de variante à cet algorithme. D'un point de vue des résultats, les réseaux à rétro-propagation sont assez efficaces, dès lors qu'ils ont été bien entraînés et qu'on leurs présente en entrée des vecteurs qui sont assez similaires aux vecteurs d'entrée qui ont servi à l'apprentissage.

Avec ce type de réseau, il est possible de réaliser l'apprentissage sur un jeu de données, en entrée et sortie, représentatif des situations possibles et d'obtenir une bonne approximation des données en sortie, même si l'apprentissage ne s'est pas effectué sur toute les données possibles.

Démarche Lorsque l'on veut utiliser la rétro-propagation pour résoudre un problème, il est primordial d'identifier les jeux de données que l'on a en entrée et que l'on souhaite associer en sortie. Une fois ces jeux de données identifiés, on crée un réseau et on effectue un apprentissage jusqu'à ce qu'il ait appris la relation liant les données en entrée avec les données en sortie. Durant cette phase d'apprentissage, l'algorithme de descente de gradient est appliqué (nous le verrons par la suite). Le jeu des données que l'on a précédemment identifié est découpé en 3 ensembles :

- un sert pour l'apprentissage en lui-même,
- un autre sert à la validation de l'apprentissage,
- le dernier sert à la généralisation (lorsqu'on présente des nouvelles données au réseau de neurones).

3.4.2.3 Optimisation de Levenberg-Marquardt

A - PRÉSENTATION

L'approche d'optimisation de Levenberg Marquardt est une méthode d'optimisation non linéaire utilisant la descente de gradient. Cette méthode améliore l'algorithme classique de descente du gradient (équation 3.10).

$$w_{i+1} = w_i - \mu \nabla E(w) \quad (3.10)$$

Avec : $\nabla E(w) = \frac{\partial E}{\partial w}$

En effet, l'algorithme classique n'est pas optimisé, et même s'il fonctionne relativement bien pour les modèles simples, il devient vraiment trop simpliste pour des modèles plus complexes. L'optimisation de Levenberg-Marquardt permet de réduire significativement le temps de convergence de l'algorithme.

B - PRINCIPE

L'algorithme de Levenberg-Marquardt consiste à résoudre l'équation :

$$(J^t + \lambda I)\sigma = J^t E \quad (3.11)$$

Dans laquelle :

- J est la matrice jacobienne du système,
- λ est le facteur d'amortissement,

δ est le vecteur de mise à jour des poids, c'est-à-dire ce que nous cherchons à déterminer,

E est le vecteur erreur contenant toutes les erreurs en sortie pour chaque vecteur d'entrée utilisé lors de l'apprentissage du réseau de neurones.

L'équation 3.12 représente la matrice jacobienne. Dans le cas des réseaux de neurones, elle représente le système. C'est une matrice de dimension N par W , où N est le nombre d'entrées dans la base d'apprentissage et W est le nombre total de paramètres (poids et biais) du réseau de neurones.

$$J = \begin{bmatrix} \frac{\partial F(x_1, w)}{\partial w_1} & \dots & \frac{\partial F(x_1, w)}{\partial w_W} \\ \vdots & \ddots & \vdots \\ \frac{\partial F(x_N, w)}{\partial w_1} & \dots & \frac{\partial F(x_N, w)}{\partial w_W} \end{bmatrix} \quad (3.12)$$

Avec :

$F(x_i, w)$ représentant la fonction du réseau évaluée pour le $i^{\text{ème}}$ vecteur d'entrée de la base d'apprentissage utilisant le vecteur poids w ,
 w_j étant le $j^{\text{ème}}$ élément du vecteur poids w .

δ nous informe de combien nous devons changer les poids du réseau pour arriver à une meilleure solution. $J^t J$ est une approximation de la matrice hessienne.

La contribution principale de Levenberg, est l'introduction du facteur d'amortissement λ . Cette valeur est additionnée à chaque élément de la diagonale de la matrice hessienne avant que le système soit résolu par le gradient.

Puis, on résout l'équation de Levenberg-Marquardt, le plus souvent en utilisant une décomposition LU. Cependant cette méthode ne marche que dans le cas où l'approximation de la matrice hessienne n'est pas devenue singulière. Si c'est le cas, on utilise une décomposition en valeurs singulières pour résoudre l'équation.

Une fois l'équation résolue, les poids w sont mis à jour en utilisant δ et les erreurs pour chaque entrée dans la base d'apprentissage sont recalculées. Si la réduction de E est rapide, on diminue λ , l'algorithme se rapproche alors de celui de Gauss Newton, et l'itération se termine. Dans le cas contraire, ou E ne diminue pas ou peu, les nouveaux poids calculés ne sont pas pris en compte et on en recalcule de nouveau en augmentant la valeur de λ , dans ce là, on s'approche de l'algorithme de descente du gradient.

Généralement, on initialise λ à 0.1 lors de la première itération et les diminutions ou augmentations de λ se font respectivement par division ou multiplication par un facteur v , le plus souvent égal à 10.

C - ALGORITHME

Dans l'algorithme de Levenberg-Marquardt, chaque itération lors de l'apprentissage (appelé aussi epoch) se déroule de la manière suivante :

1. Calculer la jacobienne
2. Calculer l'erreur du gradient
 $g = J^t \hat{E}$

3. Approximer la jacobienne en utilisant le produit vectoriel de la jacobienne
 $H = J^t J$
4. Résoudre : $(H + \lambda I)\delta = g$ to find δ
5. Mettre à jour les poids du réseau w en utilisant δ
6. Recalculer l'erreur quadratique moyenne en utilisant les nouveaux poids
 - Si l'erreur quadratique moyenne n'a pas diminué, alors on ignore les nouveaux poids, on augmente λ en le multipliant par v et on revient à l'étape 4.
 - Sinon on diminue λ en le divisant par v et on arrête l'itération.

L'algorithme s'arrête en fonction des critères que l'on fixe. Par exemple, lorsque λ est trop élevé ou que l'erreur ne diminue plus ou que très faiblement.

3.4.2.4 Les séparateurs à vaste marge (SVM)

A - PRÉSENTATION

Un séparateur à vaste marge, ou SVM, (Support Vector Machine en anglais) est une méthode mathématique proposée par Vapnik au début des années 90 [Vapnik, 2000]. Cette méthode mathématique est basée sur la théorie statistique de l'apprentissage ainsi que sur la minimisation structurelle du risque. Employés originellement pour faire de la classification, depuis quelques années, les SVMs commencent à être utilisés pour réaliser de l'identification de systèmes ou comme méthodes d'apprentissage ou encore comme méthode de régression. On les appelle dans ce cas : SVR (Support Vector Regression).

Le principal avantage des SVMs est que cette méthode permet de très bonnes performances en généralisation. Cependant ces hautes performances en généralisation se paient par des conditions d'utilisation assez contraignantes. En effet, dans la plupart des cas, la puissance des SVMs ne peut se révéler que dans les systèmes avec de petits jeux de données et permettant un minima local.

Le principe des SVMs est le suivant : à partir d'un ensemble de données appartenant à un espace, on détermine un nouvel espace de représentation de données, qui est généralement de dimension supérieure, afin de travailler aisément dans celui-ci. Une fois les traitements effectués, on repasse dans l'espace de représentation originel des données, tout en ayant préservé les traitements effectués sur les données. Nous nous intéressons seulement aux SVRs dans cette partie.

B - MODÈLE MATHÉMATIQUE DES SVRS

Les SVRs sont les SVMs appliqués au problème de la régression. La présentation suivante, de la méthode mathématique des SVRs, a été réalisée à partir de celle faite par [Adachi and Ogawa, 2002]. Tout d'abord rappelons qu'un problème de régression consiste en l'estimation d'un vecteur de paramètres $\omega = [\omega_1, \omega_2, \dots, \omega_m]^T$ d'une fonction inconnue $f(x_i, \omega)$ basée sur les données observées y_i :

$$y_i = f(x_i, \omega) + e_i, \quad i = 1, 2, \dots, N \quad (3.13)$$

Avec :

x_i les données en entrée,

e_i les erreurs,
 N le nombre de données.

Le couple (x_i, y_i) forme ce que l'on appelle la base d'exemple, c'est à dire un ensemble de données en entrées associées à leurs sorties respectives.

La fonction de régression aura alors la forme suivante :

$$f(x_i, w) = \sum_{j=1}^m \omega_j g_j(x_i) \quad (3.14)$$

Avec:

g_i exprimant un ensemble de transformations non linéaires.

Afin d'utiliser les SVMs dans des problèmes de régression, il faut définir une nouvelle fonction de perte, appelée ε -insensible, définie par :

$$L_\varepsilon(y_i, f(x_i, w)) = \begin{cases} 0, & \text{si } |y_i - f(x_i, w)| \leq \varepsilon \\ |y_i - f(x_i, w)| - \varepsilon, & \text{sinon} \end{cases} \quad (3.15)$$

Avec:

ε étant un paramètre permettant de contrôler la largeur de la marge et qui fixe le nombre de séparateurs.

En utilisant la définition du risque empirique de [Vapnik](#) : S_k [[Vapnik, 2000](#)] tel que :

$$S_k = \sum_{j=1}^m \omega_j g_j(x) \quad \text{Soumis à la contrainte : } (w \cdot w) \leq c_k \quad (3.16)$$

On peut donc poser le premier problème comme étant la détermination du vecteur w qui minimise le risque empirique défini par la fonction R_{emp} , soit :

$$\text{Min } R_{\text{emp}}(w) = \frac{1}{N} \sum_{i=1}^N L_\varepsilon \left(y_i, \sum_{j=1}^m \omega_j g_j(x_i) \right) \quad (3.17)$$

Avec comme contrainte : $(w \cdot w) \leq c_k$

Ce problème peut être transformé en ce second problème, en introduisant des multiplicateurs de Lagrange pour chaque contrainte, dans lequel on doit trouver les paramètres α_i^* et β_i^* qui maximiserons la fonction $Q(\alpha, \beta)$:

$$\begin{aligned} \text{Max } Q(\alpha, \beta) = & -\varepsilon \sum_{i=1}^N (\alpha_i + \beta_i) + \sum_{i=1}^N y_i (\alpha_i - \beta_i) \\ & - \frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \beta_i)(\alpha_i - \beta_i) H(x_i, x_j) \end{aligned} \quad (3.18)$$

Qui sera soumise aux contraintes :

$$\sum_{i=1}^N \alpha_i = \sum_{i=1}^N \beta_i, \quad 0 \leq \alpha_i, \beta_i \leq C, \quad i = 1, \dots, N \quad (3.19)$$

Avec :

H une fonction noyau,
 ε une zone insensible,
 C un paramètre de régularisation.

Exemple de fonctions noyaux :

- Linéaire : $H(x, x') = x \cdot x'$
- Polynomial (de degrés d) : $H(x, x') = (x \cdot x')^d$ ou $(c + x \cdot x')^d$
- Gaussien (d'écart-type σ) : $H(x, x') = e^{-|x \cdot x'|^2 / \sigma}$
- Laplacien (d'écart-type σ) : $H(x, x') = e^{-\|x \cdot x'\|_1 / \sigma}$

Ce second problème peut être résolu par une méthode de programmation quadratique qui nous permettra de déterminer les solutions α_i^* et β_i^* . La fonction de régression aura alors cette forme :

$$f(x) = \sum_{i=1}^N \gamma_i^* H(x_i, x) \quad (3.20)$$

Avec :

$$\gamma_i^* = \alpha_i^* - \beta_i^*$$

REMARQUE : Par des essais successifs, en jouant sur les paramètres ε et C, on intervient sur le bruit et sur le lissage de la fonction de régression. Ces paramètres sont choisis *a priori*, de manière empirique.

C - UTILISATION DES SVRS POUR L'IDENTIFICATION

Zhang et al, ainsi que An et al, ont démontré l'intérêt de l'utilisation des SVMs pour l'identification de systèmes non linéaires face à l'utilisation des réseaux de neurones pour l'identification [Zhang et al, 2004; An et al, 2007]. Ils ont conclu dans leurs études que le temps d'apprentissage et la généralisation ainsi que l'approximation, sont meilleurs avec les SVMs qu'avec les réseaux de neurones. Cependant ces résultats ont été obtenus sur un système monodimensionnel et on ne peut donc pas prendre leurs conclusions comme pertinentes pour tous les systèmes existants, en particuliers pour les systèmes multidimensionnels.

Adachi and Ogawa ont proposé une méthode d'identification basée sur les SVRs pour des modèles de régression linéaire [Adachi and Ogawa, 2002]. Ils ont démontré que leur méthode disposait de bonnes performances pour l'apprentissage pour des ensembles de données finis. Cependant il ne faut pas oublier que leur méthode ne s'applique qu'aux systèmes linéaires ce qui est rarement le cas.

Zhao and Han ont, quant à eux, proposé l'utilisation des SSVRs (Smooth Support Vector Regression) afin de réduire la complexité arithmétique inhérente aux SVRs [Zhao and Han, 2007]. Cette réduction de la complexité permet d'appliquer les SVRs sur des bases de données plus grandes, car rappelons le, le temps d'apprentissage augmente

fortement avec l'augmentation du nombre d'échantillons. Ils ont obtenus plus rapidement des meilleurs résultats avec leur méthode en l'appliquant sur un modèle d'état de turbine.

3.5 LA CARACTÉRISATION D'UN SYSTÈME DE TRACKING

3.5.1 *Evaluation des distorsions d'un système de tracking*

Les distorsions d'un système de tracking peuvent être représentées de différentes manières. Il ne faut pas oublier que ces distorsions sont fonction des positions relevées par le système de tracking et des positions théoriques auxquelles l'entité doit être située. La manière la plus évidente de représenter cette fonction est le vecteur d'erreur sur la position. Ce vecteur erreur \vec{e} est défini par :

$$\vec{e}_i = \vec{P}_{Ti} - \vec{P}_{Ri} \quad (3.21)$$

Avec:

- P_T Position théorique à laquelle l'entité est censée être située,
- P_R Position de l'entité relevée par le système de tracking.

A partir de ce vecteur, on peut déterminer l'erreur absolue e_{Abs} à l'instant i :

$$e_{Absi} = \|\vec{e}_i\| = \|\vec{P}_{Ti} - \vec{P}_{Ri}\| \quad (3.22)$$

En étudiant les valeurs caractéristiques de ces erreurs, comme la moyenne, la variance et l'écart-type ainsi que les valeurs maximales on peut estimer la précision, la confiance ou bien la dispersion du système de tracking.

3.5.2 *Quelques moyens de représentation des distorsions d'un système de tracking*

Afin de pouvoir évaluer la meilleure méthode de calibration et caractériser un système de tracking, il est important de pouvoir visualiser les distorsions dans l'espace de travail afin que l'expert puisse juger de la méthode la plus appropriée pour la calibration. Cela peut paraître trivial d'étudier ces méthodes de représentation, mais la vérité est qu'il est très complexe de caractériser un système de tracking 3D et il est donc par conséquent très difficile de choisir une bonne méthode de calibration/correction ainsi que de la justifier. Une des possibilités de représentation des distorsions, dans l'espace de travail du système de tracking, est de tracer les vecteurs erreurs en 3D, en utilisant un logiciel mathématique, comme MATLAB® [MathWorks, 2010] par exemple. Ghazisaedy and Adamczyk ont, dans leurs travaux de recherche sur la calibration ultrasonique d'un système de tracking magnétique [Ghazisaedy and Adamczyk, 1995], utilisé ce type de représentation (voir figure 62).

Une autre possibilité, utilisée dans de nombreux travaux de recherche, consiste à représenter l'erreur sur la position en fonction de la distance des sources de mesure du système (figure 64 et figure 63). L'avantage de cette représentation est qu'elle permet

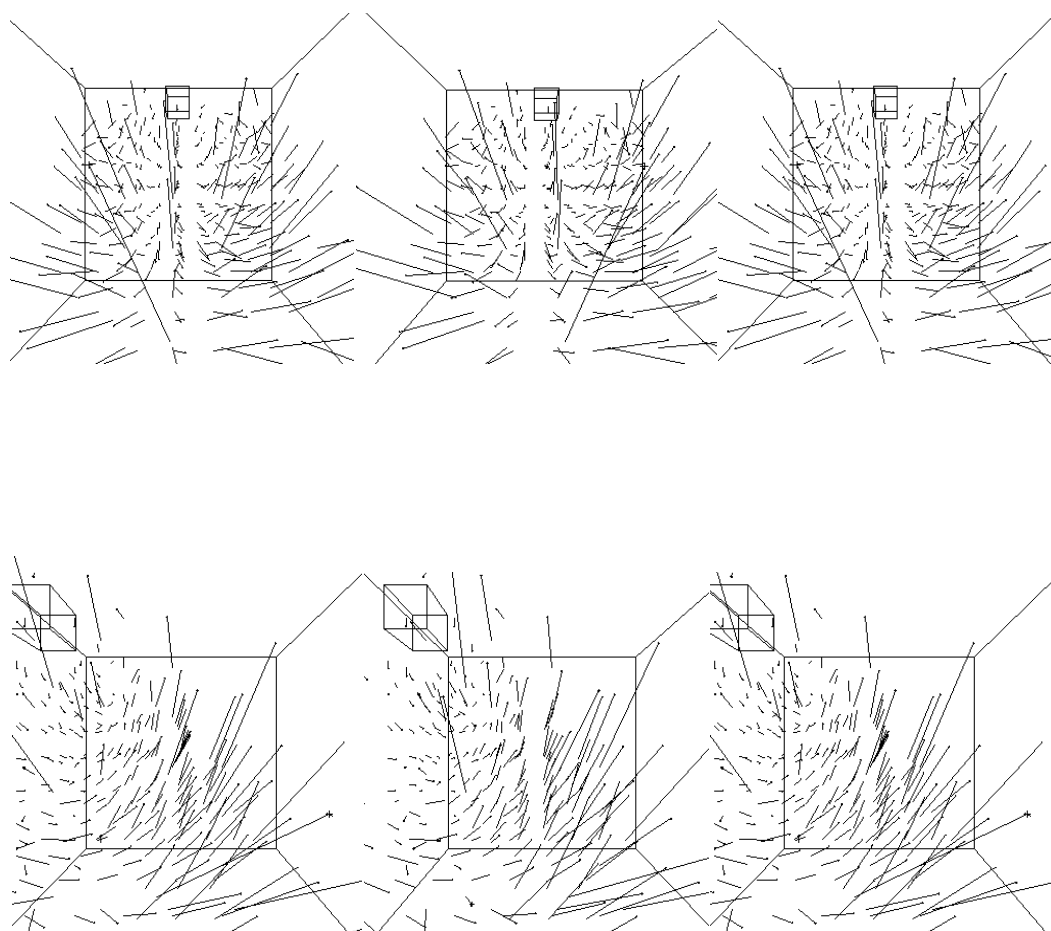


Figure 62: Représentation des distorsions sous la forme de vecteurs 3D dans l'espace de travail adapté de [Ghazisaedy and Adamczyk, 1995].

généralement de démontrer que l'erreur augmente en fonction de la distance à laquelle on est située des sources de tracking. De plus, elle permet de montrer si le système de tracking est linéaire ou non.

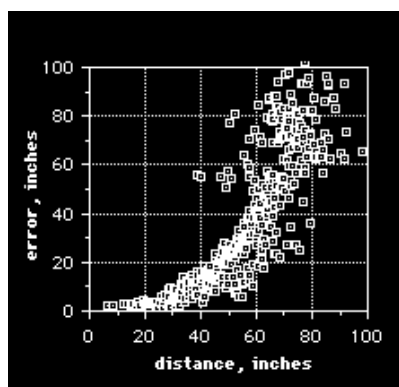


Figure 63: Représentation de l'erreur en fonction de la distance des sources dans les travaux de Bryson [Bryson, 1992].

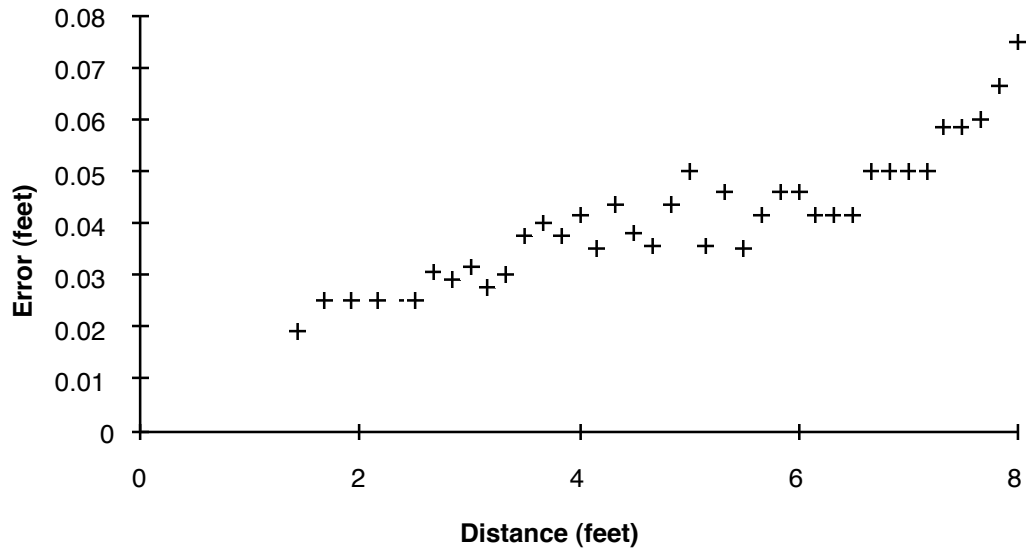


Figure 64: Représentation de l'erreur en fonction de la distance des sources dans les travaux de Ghazisaedy and Adamczyk [Ghazisaedy and Adamczyk, 1995].

On peut également superposer les données relevées par le système de tracking sur les données théoriques auxquelles elles devraient être égales. On peut voir ce type de représentation sur la figure 65 provenant des travaux de Ikits et al [Ikits et al, 2003].

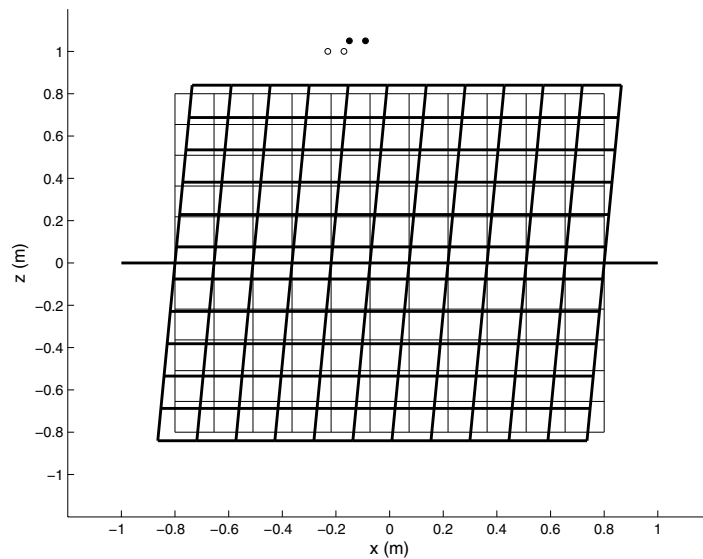


Figure 65: Superposition des positions relevées avant et après calibration [Ikits et al, 2003].

Ikits et al ont également représenté les erreurs sur la calibration pour chacun des points ayant servi à la calibration [Ikits et al, 2003]. L'avantage de cette représentation (voir figure 66) est qu'elle permet de distinguer la répartition des erreurs sur la grille de calibration. Mais on peut également tracer l'histogramme des erreurs dans le volume de travail.

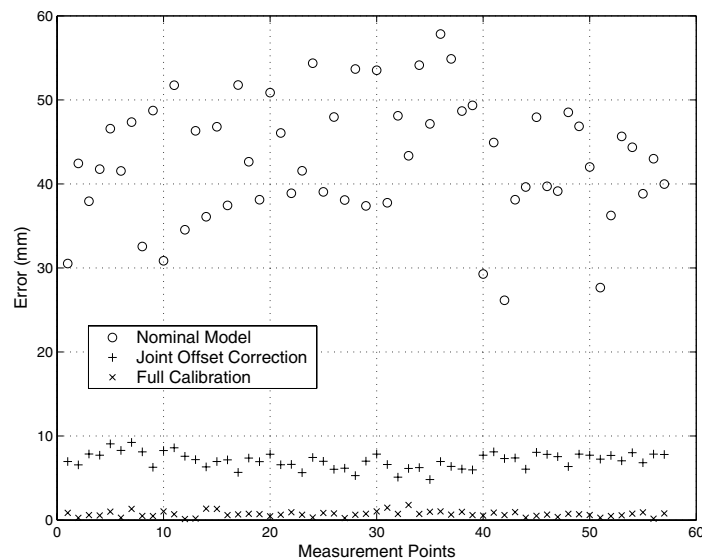


Figure 66: Représentation des erreurs pour chacun des points de la grille de calibration [Ikits et al, 2003]

3.6 QUELQUES EXEMPLES DE CALIBRATION DE SYSTÈMES DE TRACKING

3.6.1 La correction de distorsions statiques sur les systèmes de tracking 3D

Bryson, dans un rapport technique datant de 1994, a étudié la calibration et les effets de la présence de distorsions statiques dans les systèmes de tracking 3D [Bryson, 1992]. Dans ce rapport il propose une méthode afin d'évaluer différentes méthodes de calibration ainsi que le résultat de différentes méthodes de calibration sur un système de tracking magnétique.

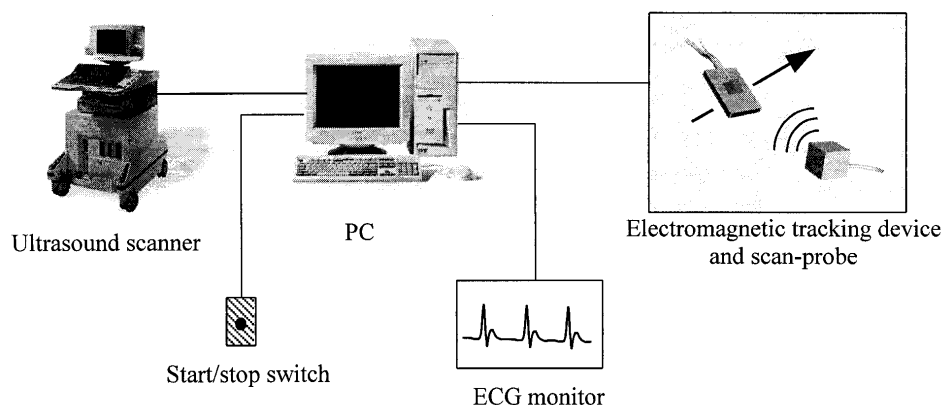


Figure 67: Le dispositif de tracking hybride mis en place par Barratt et al [Barratt et al, 2001].

Barratt et al a proposé un système de tracking basé sur les ultrasons s'auto-calibrant afin d'améliorer sa précision [Barratt et al, 2001]. Ce système est représenté figure 67. Pour cela il utilise aussi un système de tracking optique et un algorithme de recalage

exploitant une méthode basée sur l'ajustement par les moindres carrés proposée par Fitzgibbon [Fitzgibbon, 2003].

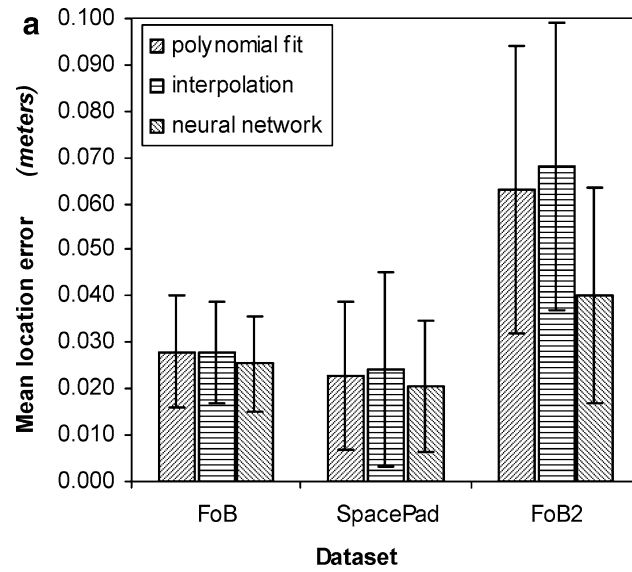


Figure 68: Résultats obtenus par Kindratenko and Sherman via l'utilisation d'un réseau de neurones par rapport à d'autre méthodes [Kindratenko and Sherman, 2005].

Kindratenko and Sherman ont réalisé une technique de calibration pour les systèmes de tracking magnétique, basée sur les réseaux de neurones [Kindratenko and Sherman, 2005] afin de corriger les erreurs de positionnement et d'orientation dues aux distorsions du champ magnétique induit par l'environnement dans lequel est situé le système. Dans leurs recherches, ils ont justifié l'utilisation de chaque type de réseaux de neurones ainsi que le nombre optimal de neurones dans la couche cachée. Ils arrivent à obtenir des résultats très intéressants face à l'utilisation d'autre méthodes, telles que l'interpolation polynomiale ou la simple interpolation linéaire (figure 68). Le réseau de neurones utilisé s'avère très efficace.

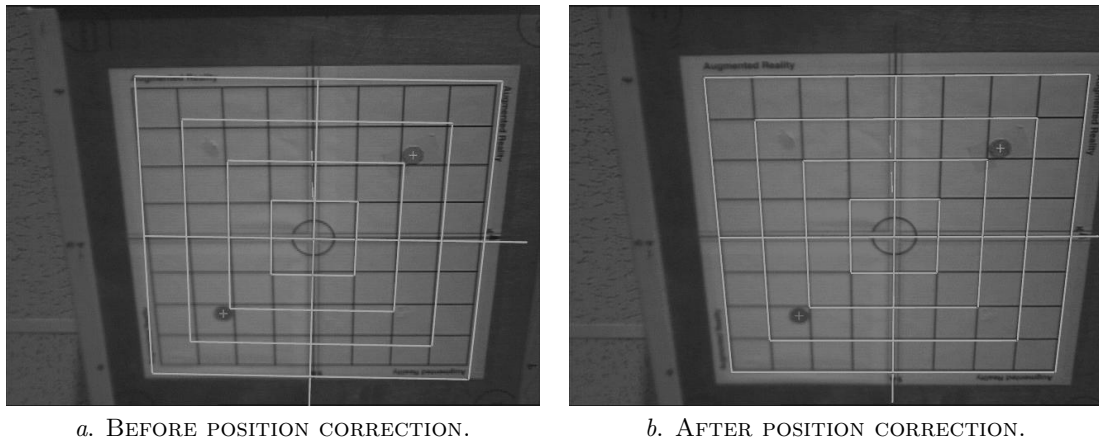


Figure 69: Avant et après correction du tracking par la méthode proposée par Livingston [Livingston, 1997].

Livingston a utilisé une grille de correspondance afin de calibrer un système de tracking magnétique [Livingston, 1997] en vue de l'utiliser dans un environnement virtuel

(figure 69). Il a, de plus, restreint la zone de travail du système afin d'en améliorer sa précision. Il arrive à une réduction de 79% de l'erreur sur la position. Afin de tester les effets de sa calibration il a comparé les résultats avec une table de calibration et sans table de calibration, mais à l'extérieur afin d'être dans un environnement sans interférence.

3.6.2 Système de tracking hybride

Birkfellner et al ont développé un système de tracking hybride basé sur un tracking électromagnétique et optique, destiné à être utilisé dans la chirurgie [Birkfellner et al, 1998]. Le but d'un tel système est de garantir la continuité du tracking ainsi que de permettre de pallier les défauts des deux systèmes :

- Le tracking magnétique souffre de distorsions : on les corrige en le calibrant à l'aide du tracking optique,
- Le tracking optique est précis mais il souffre de perte de la continuité du tracking due aux fréquentes occultations : dans ces situations le tracking magnétique peut le suppléer.

Le logigramme représenté dans la figure 70 explique le basculement du système de tracking optique (OTS pour Optical Tracking System) vers le système de tracking magnétique (EMTS pour Electro-Magnetic Tracking System). Des tests sont effectués pour savoir si chacun des systèmes est disponible et fonctionnel. Dans le cas où le système de tracking magnétique est utilisé, on transforme ses données dans l'espace de coordonnées du système de tracking optique. Dans le cas échéant, le chirurgien est prévenu afin qu'il suspende son activité.

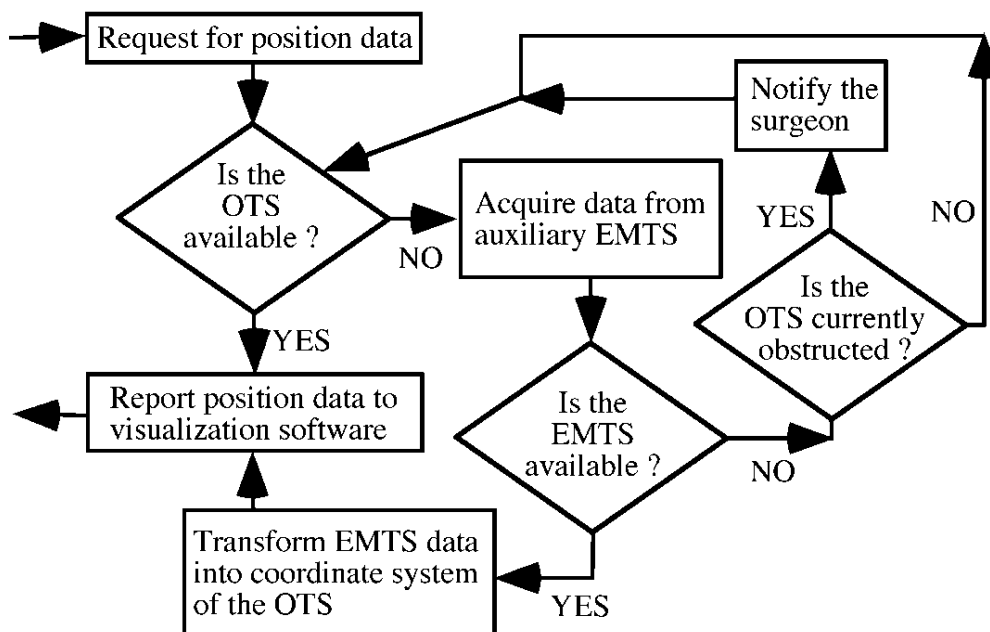


Figure 70: Logigramme du système de tracking hybride proposé par Birkfellner et al [Birkfellner et al, 1998].

3.7 CONCLUSION

Nous avons pu voir dans ce chapitre les différentes approches afin de calibrer un système de tracking 3D. Dans un premier temps nous avons défini ce qu'est une calibration. Puis, nous avons évoqué les difficultés rencontrées, d'une part pour caractériser ces systèmes et d'autre part, de proposer un protocole de calibration approprié. Enfin, nous avons vu un bref aperçu des différentes méthodes mathématiques permettant de procéder à la calibration de ces systèmes.

Je tiens à signaler que les travaux dont j'ai parlé dans cette section, sont ceux que j'ai sélectionnés pour leurs similarités avec ceux de réalité virtuelle, tant au niveau des périphériques utilisés que pour leurs applications. Cependant, dans le domaine de la réalité augmentée, un très grand nombre de travaux traitent de ce sujet, mais avec la plupart du temps des capteurs marchant en extérieur (GPS) ou bien des capteurs généralement peu utilisés en réalité virtuelle (centrale inertielle). Mais certaines approches mathématiques sont intéressantes et peuvent être envisagées en réalité virtuelle. On pourra aller consulter la thèse de [Maidi, 2007] ou bien l'état de l'art de [Zendjebil et al, 2009] qui recensent de nombreuses méthodes pour améliorer le suivi ou la localisation d'objets.

Suite à cette étude, nous pouvons en conclure, qu'il n'existe pas de méthodes génériques pour calibrer un système de tracking 3D, et qu'il faut procéder à une caractérisation de chacun des systèmes avant même de décider quelle approche adopter. Cette caractérisation nous permet de mettre en avant les avantages et les inconvénients du système de tracking.

En fonction des résultats de cette caractérisation, une personne, que l'on qualifiera d'expert, choisira une ou plusieurs approches pour procéder à la meilleure calibration possible. Chaque méthode a ses avantages et ses inconvénients. De plus certaines méthodes ne peuvent pas être appliquées en pratique, soit pour des raisons de temps de réponse, soit pour des raisons de consommation de ressources (puissance de calcul et mémoire).

L'ensemble de ce chapitre démontre bien la difficulté à utiliser les systèmes de tracking de manière fiable et surtout le choix d'une méthode de calibration appropriée, vu que celle-ci est entièrement dépendante de la technologie utilisée, l'environnement dans lequel le système est situé, ainsi que ses conditions d'utilisation.

Partie II

CONTRIBUTIONS

PROPOSITION D'UN SYSTÈME D'INTERACTION 3D ADAPTATIVE

4.1 INTRODUCTION

Le but de cette thèse est d'apporter une contribution aux systèmes d'interaction 3D multi-capteurs en réalité virtuelle en les rendant plus adaptatifs aux faits et gestes de l'utilisateur. L'apport de l'adaptativité dans ces systèmes est principalement d'interpréter les faits et gestes de l'utilisateur afin que l'interaction 3D paraisse la plus naturelle possible pour l'utilisateur.

Nous définissons une interaction 3D naturelle telle que :

- L'interaction 3D soit continue,
- La finalité d'une action réelle de l'utilisateur afin d'agir sur l'environnement virtuel, sous la forme d'une action virtuelle soit perçue comme logique pour l'utilisateur.

Cette contribution se présente donc sous la forme d'un concept qui est axé sur la continuité des données provenant des différents périphériques ainsi que celle de l'interaction 3D, que cela soit au niveau de la réalisation de la tâche (navigation, sélection ou manipulation) ou le passage de l'une vers une autre (contrôle d'application). La continuité de l'interaction 3D est un critère très important dans l'ergonomie qu'éprouvera l'utilisateur lorsqu'il accomplira des tâches au sein de l'environnement virtuel. En effet, nous nous sommes fixés comme objectif de rendre l'interaction 3D la plus naturelle possible pour un utilisateur lambda, nous avons donc dû réfléchir à un système capable de s'adapter automatiquement au comportement de l'utilisateur ainsi que de proposer une technique d'interaction 3D innovante. Pour cela, nous nous sommes intéressés à l'étude des architectures multi-capteurs et multimodales (chapitre 1.5) ainsi qu'aux capteurs d'interaction en réalité virtuelle (chapitre 2). En maîtrisant toutes ces facettes, nous avons été à même de proposer un système d'interaction 3D adaptatif accordant à l'utilisateur la possibilité d'interagir naturellement avec l'environnement virtuel.

Enfin, cette contribution est aussi le résultat fructueux d'une collaboration étroite entre deux domaines qui ne s'associent que trop rarement : celui de la réalité virtuelle ainsi que celui du traitement et de la fusion de données.

4.2 DÉMARCHE SCIENTIFIQUE:

En partant de la volonté de préserver le continuum de l'interaction 3D, nous avons défini en premier lieu la nécessité de définir une technique supportant les différentes tâches de l'interaction 3D. Cette technique nécessite d'être couplée à un système permettant d'estimer la tâche de l'interaction 3D que souhaite réaliser l'utilisateur. De plus cette technique se veut d'être compatible avec le plus grand nombre de périphériques

de tracking possible. Nous l'avons donc développée de manière à ce qu'elle ne nécessite qu'un seul vecteur position. Le problème avec cette approche, est que ce vecteur position doit être continu, le système de tracking doit donc être robuste et fournir ce vecteur position en continu, sous peine de rompre la continuité de l'interaction 3D. D'autre part, il est souvent nécessaire d'améliorer le système de tracking, tant au niveau de la précision, qu'au niveau de la continuité des données. On aura donc besoin d'un composant capable d'effectuer des traitements sur les données ou bien de combiner plusieurs capteurs. Dans le cas où on utilisera plusieurs capteurs, il sera primordial de les calibrer l'un à l'autre de manière à ce qu'ils soient synchrones et qu'ils travaillent dans le même espace de données. Le diagramme représenté en figure 71 représente le fruit de notre réflexion.

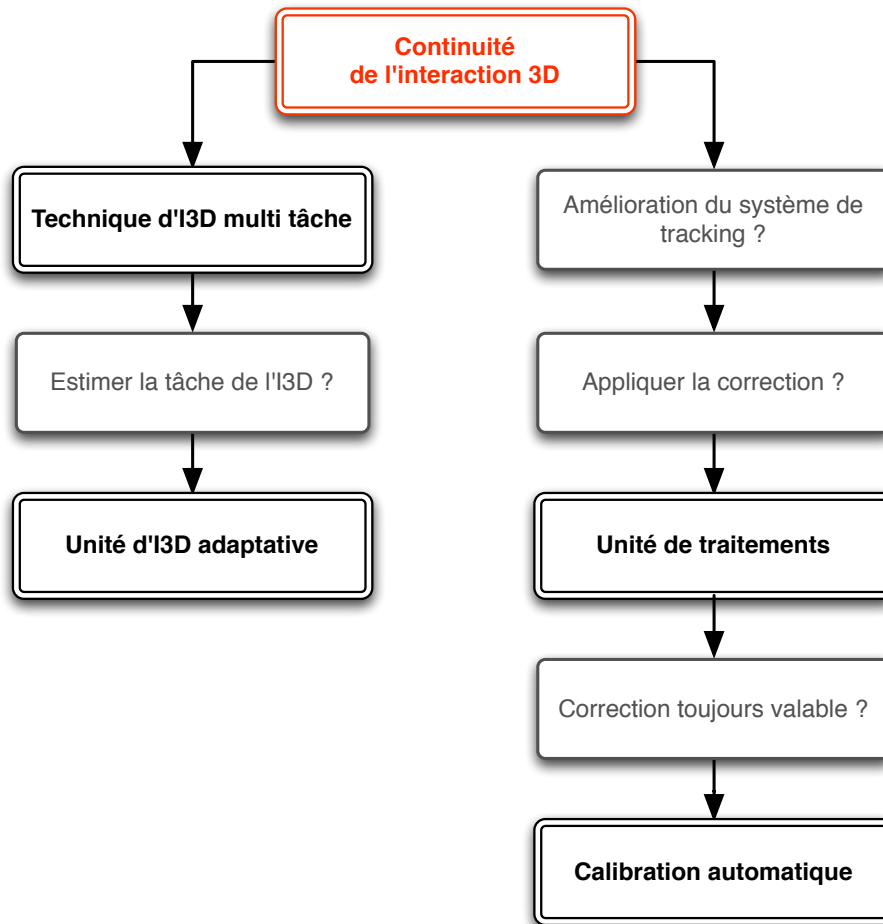


Figure 71: Réflexion sur les différents composants nécessaires pour répondre à l'objectif.

4.3 ARCHITECTURE PROPOSÉE

La figure 72 représente l'architecture que nous proposons, en réponse à la réflexion précédemment réalisée. L'architecture est basée sur trois composants principaux, correspondant à trois aspects différents d'un système de réalité virtuelle :

- l'Unité de Correction et Combinaison des Données, appelé *UCCD*, s'occupant plutôt de la partie interaction matérielle et gestion de périphériques,

- l'Unité d'Interaction 3D Adaptative, appelé *UI3D-A*, s'occupant de la partie interaction 3D et "intelligence" du système,
- la technique d'interaction 3D, permettant l'interaction 3D entre l'utilisateur et l'environnement virtuel.

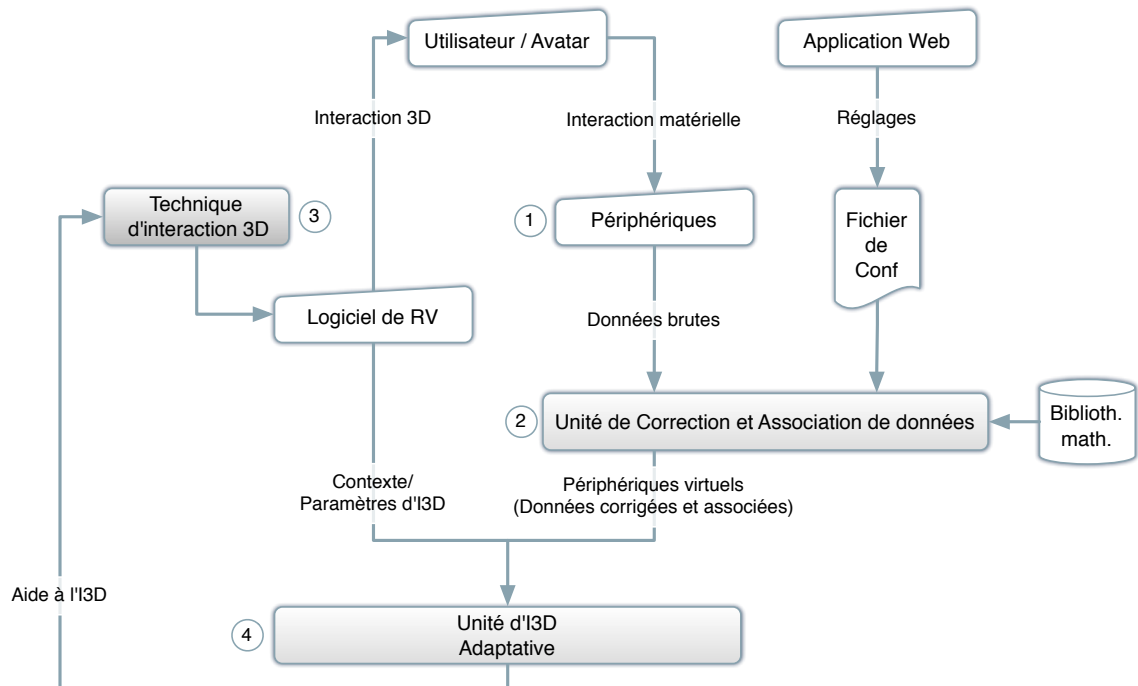


Figure 72: Cycles de l'architecture.

On distingue quatre étapes successives dans ce modèle.

1. L'utilisateur interagit avec le matériel. A ce niveau, la couche matérielle se contente juste de rendre disponible les données brutes provenant des différents périphériques.
2. Un premier traitement a lieu, qui consiste à effectuer les premières corrections éventuelles des différents périphériques. Les données une fois corrigées sont mises à disposition sur le réseau. Puis, le système effectue divers traitements sur les données. C'est ici que les données peuvent être associées, combinées ou transformées afin de réaliser des périphériques virtuels qui sont le résultat de multiples traitements sur des périphériques réels. Ensuite, ces données sont normalisées et redéfinies dans un espace de représentation commun.
3. Enfin, l'interaction 3D a lieu entre l'utilisateur, représenté dans l'environnement virtuel par son avatar et le logiciel de réalité virtuelle.
4. Une dernière étape consiste à ce que le système vienne assister l'utilisateur sur l'interaction 3D de manière non perceptible afin de l'aider dans la réalisation de tâches dans l'environnement virtuel.

4.4 LA TECHNIQUE D'INTERACTION 3D : FLY-OVER

4.4.1 Introduction

Cette partie est une contribution aux systèmes d'interaction 3D en réalité virtuelle. Elle leur apporte une nouvelle technique d'interaction qui s'adapte aux comportements de l'utilisateur et qui est compatible avec n'importe quel périphérique. Cette technique fait partie intégrante du système d'interaction 3D adaptative que nous proposons, nous l'avons nommée : Fly-Over.

Un des aspects important d'un système d'interaction 3D adaptative est qu'il doit préserver la continuité de l'interaction 3D, tout cela afin de garantir à l'utilisateur la meilleure expérience utilisateur possible. Cette continuité s'appuie sur une gestion efficace et performante des périphériques ainsi que par une bonne interface utilisateur. L'approche que nous avons choisie suit deux objectifs principaux :

1. Cette technique d'interaction doit être le plus générique possible, que ce soit d'un point de vue de son utilisation avec n'importe quel type de périphérique de pointage, ou bien par rapport à son utilisation dans les quatre tâches de l'interaction 3D qui doit obéir au même principe afin de ne pas perturber l'utilisateur lors du passage d'une tâche à une autre.
2. Elle doit permettre à des utilisateurs experts et non-experts de réaliser les tâches courantes de l'interaction 3D avec un faible temps d'apprentissage.

Pour la gestion efficace et performante des périphériques nous nous sommes appuyés sur l'unité de correction et de combinaison des données que nous avons conçue. Je vous expliquerai dans la section 4.6.5 la relation qu'il existe entre celui-ci et notre technique d'interaction 3D. Quant à la conception d'une bonne interface utilisateur, elle s'est architecturée autour de cette généricité dans l'utilisation et dans son comportement face à différents périphériques qu'ils soient à 6, 3 ou 2 degrés de liberté. Nous avons donc dû chercher une nouvelle métaphore pour caractériser notre technique d'interaction 3D. Cela ne veut pas pour autant dire que nous sommes repartis de zéro, mais plutôt que nous avons proposé une base innovante sur laquelle nous avons greffé des techniques ou des principes existants. Cependant, l'ensemble forme un tout cohérent et homogène garantissant ainsi une utilisation simple et naturelle pour l'utilisateur et garantissant la continuité de l'interaction 3D.

4.4.2 Spécification du modèle générique de Fly-Over

D - CONTRAINTES

Après études des techniques d'interaction 3D existantes et des limites des architectures multi-capteurs, nous avons choisi de proposer une base pour Fly-Over qui doit respecter trois contraintes, afin que cette nouvelle technique puisse répondre aux objectifs fixés. Voici ces trois contraintes :

1. Être compatible avec tous les périphériques de pointage 2D, 3D et 6D, étant capable de retourner une position 2D ou 3D voire 6D de l'utilisateur ou de l'effecteur que manipule l'utilisateur.

2. Maintenir la même logique d'utilisation pour l'utilisateur quel que soit le périphérique utilisé et quel que soit la tâche de l'interaction 3D à effectuer.
3. Garantir à tout niveau d'expertise de l'utilisateur une courte durée d'apprentissage.

Afin de respecter ces contraintes, nous avons proposé un modèle générique basé sur deux idées. la première étant que toutes les tâches de l'interaction 3D vont être réduites en une simple tâche de pointage. La seconde est de découper l'espace à 6 dimensions autour de l'utilisateur en plusieurs sous-espaces dans chacun desquels pourront être effectués différentes tâches. Avec ces deux principes, nous respectons toutes les contraintes précédemment évoquées.

E - MODÈLE MATHÉMATIQUE

L'interaction entre un utilisateur dans le monde réel avec un monde virtuel consiste en une boucle dans laquelle l'utilisateur manipule un ensemble de périphériques qui envoient des informations et ces informations agissent sur l'environnement virtuel. A chaque instant, les manipulations que l'utilisateur exerce sur un périphérique de pointage afin d'agir sur l'environnement virtuel peut être décrit comme un vecteur V de dimension $m \leq 3$ dans un espace $\Omega \subset \mathbb{R}^m$.

Là où la plupart des techniques d'interaction 3D transforment directement la valeur du vecteur V en une action sur l'environnement virtuel, nous préférons utiliser un espace intermédiaire $Z \subset \mathbb{R}^m$ dans lequel la tâche de pointage a lieu. En fonction des positions données par V dans Z , différentes actions seront exécutées dans l'environnement virtuel. En d'autres termes, l'idée est de transformer une tâche complexe invoquant m dimensions en des tâches invoquant un nombre de dimensions inférieur ou égal à 3.

Nous appelons f , la transformation qui permet de passer de la position dans le référentiel du système de tracking : $V \in \Omega$ à une position exprimée dans un nouveau référentiel $P \in Z$.

$$P = f(V) \quad (4.1)$$

Afin de décrire les différentes tâches possibles, nous avons subdivisé Z en plusieurs sous-espaces correspondant chacun à une tâche.

$$Z = \cup Z_i \quad i \in \{1, 2, \dots, n\} \quad (4.2)$$

Chaque vecteur position $P \in Z$ peut être associé à un ensemble de couples : action (F_i) et paramètre de l'action (I_i), à l'aide de la fonction g :

$$g(P) = (F_i(I_1, I_2, \dots, I_m), \quad i \in \{1, 2, \dots, q\} \quad (4.3)$$

Avec :

F_i , une fonction définissant une action élémentaire (ex : translation, rotation).

I_j , un paramètre optionnel qui peut être utilisé pour agir sur la fonction F_i définissant l'action élémentaire associée (ex : gain).

Tout cela peut être résumé par :

$$\forall V \in \Omega \quad h(V) = g \circ f(V) = \{(F_i, I_i)\} \quad (4.4)$$

4.4.3 Fly-Over pour la tâche de navigation : Fly-Over-N

4.4.3.1 Premier modèle

Fly-Over-N est le modèle de Fly-Over dédié à la tâche de navigation. Des constats généraux basés sur l'observation de tâches de navigation en environnement virtuel, nous ont permis de déterminer le nombre n de sous-espace Z_i et leurs dimensions associées. Les constats sont les suivants :

- Gérer simultanément les translations et les rotations dans un environnement virtuel qui est rendu dans une plateforme (semi-)immersive, peuvent déclencher des malaises ou troubler l'utilisateur.
- Le constat général, est que les utilisateurs ont une tendance à d'abord s'orienter de manière à avoir l'objet ou la zone vers laquelle ils souhaitent aller en face d'eux, puis de traduire dans cette direction.

L'idée est donc de découpler en partie l'espace de navigation, qui est normalement constitué en 6 degrés de liberté, en deux sous-espaces possédant chacun 3 degrés de liberté. Nous avons choisi de modéliser ces 2 sous-espaces à l'aide de 2 sphères concentriques : la première sphère représentant le premier sous-espace Z_1 et le volume entre la première et la seconde sphère représentant le second sous-espace Z_2 . Le premier sous-espace Z_1 sera celui dans lequel on effectuera des rotations et le second, Z_2 , sera celui dans lequel on effectuera des translations. L'ensemble de ces zones formant le blob d'interaction 3D de Fly-Over. Ce premier modèle est celui représenté dans la figure 73.

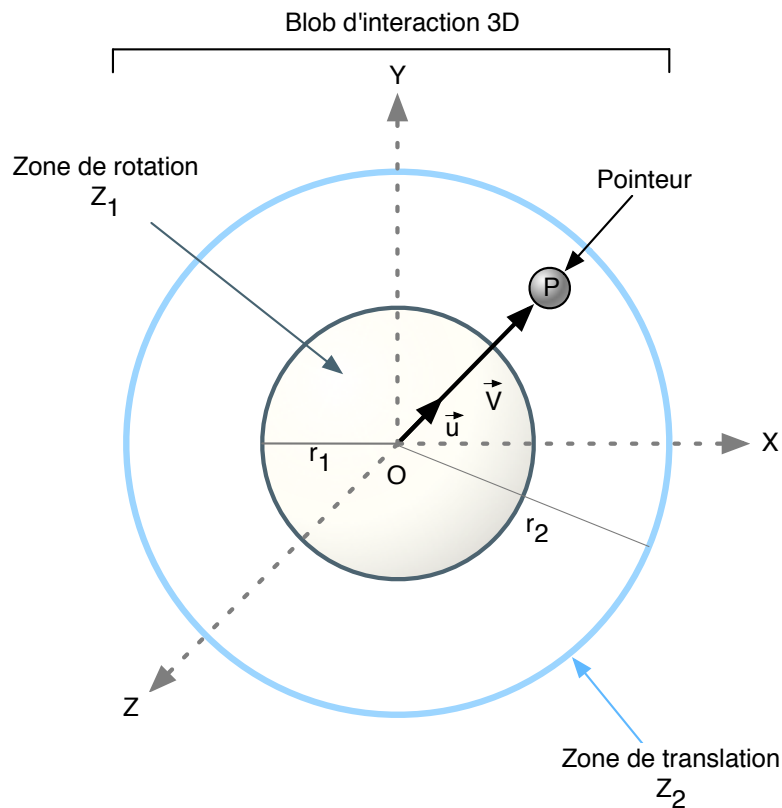


Figure 73: Modèle simplifié de Fly-Over - N.

Ainsi dans un espace limité autour de la main de l'utilisateur, celui-ci peut effectuer des rotations de son point de vue. Lorsqu'il commencera à tendre son bras, les transformations appliquées au point de vue seront transformées en translation.

A - ACTIONS ET PARAMÈTRES

Dans ce premier modèle dédié à la navigation (figure 73), les actions associées à chaque zone Z_i d'interaction sont évidentes : translation et rotation. Mais nous leurs avons également associées des paramètres.

- La translation nécessite deux paramètres : un vecteur directeur et la norme du vecteur. Le premier \vec{u} indique la direction dans laquelle translater le point de vue, le second $\|\vec{V}\|$ nous indique la vitesse de translation, ainsi nous avons donc :

$$\text{Translation : } F_2(\vec{u}, \|\vec{V}\|) \quad (4.5)$$

- La rotation consiste juste à définir la direction dans laquelle le point de vue doit regarder, le seul paramètre à connaître est : \vec{V} . On a donc :

$$\text{Rotation : } F_1(\vec{V}) \quad (4.6)$$

- Pour n'effectuer aucune transformation, il suffit de placer le pointeur dans une zone sphérique au centre du blob possédant un rayon très faible.

B - PROBLÈMES INHÉRENTS À CE MODÈLE

Cependant avec ce premier modèle, deux problèmes font leur apparition :

1. Tout d'abord, il existe une discontinuité lors du passage de la zone de rotation vers la zone de translation et vice-versa. Le passage de l'une vers l'autre se réalise de manière abrupte et peut causer un désagrément pour l'utilisateur.
2. De plus, si l'utilisateur déplace son pointeur trop rapidement dans la zone de rotation (ex : deux positions opposées) la rotation, telle qu'elle est gérée dans ce modèle, s'effectuera instantanément générant ainsi, là encore, un éventuel désagrément, et dans tous les cas, un manque de compréhension de ce qu'il vient de se passer, de la part de l'utilisateur.

Nous proposons donc une évolution à ce premier modèle.

4.4.3.2 Evolution du modèle de Fly-Over-N

Afin de résoudre les deux problèmes précédemment cités, nous avons proposé d'utiliser une zone intermédiaire qui servira de transition entre la zone de rotation et la zone de translation. Ce sera donc une zone dans laquelle le point de vue effectuera à la fois des rotations et des translations. Ainsi on lissera le passage d'une zone à une autre éliminant ainsi une quelconque gêne de l'utilisateur. Cette zone de transition Z_3 est définie comme une couronne sphérique définie par les rayons r' et r'' .

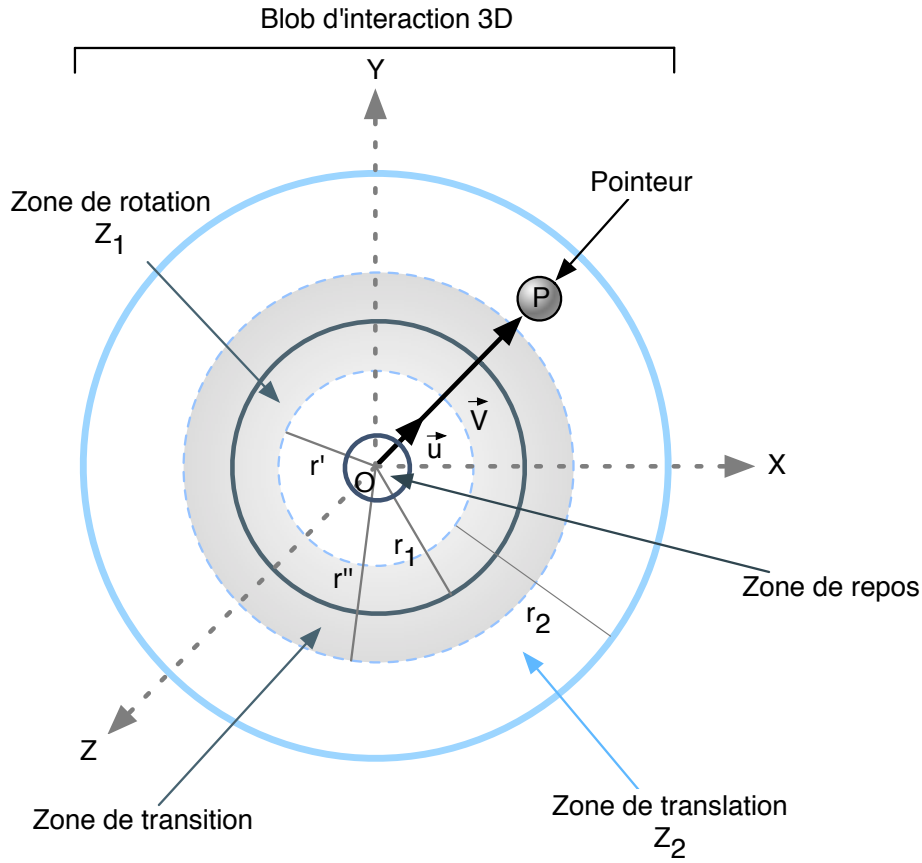


Figure 74: Evolution du modèle de Fly-Over - N avec l'ajout d'une zone de transition entre la zone de rotation et de translation.

A - AJOUT DE POIDS AUX TRANSFORMATIONS

Afin de définir la part de chacune des transformations appliquées au point de vue, nous introduisons deux coefficients pondérateurs : C_R et C_T . C_R est là pour pondérer l'amplitude des rotations (la vitesse de rotation) tandis que C_T pondère l'amplitude de la translation (la vitesse de translation). En fait, ces deux coefficients sont des poids variables et sont donc des fonctions, fonctions dépendantes de d , soit la distance séparant le centre du blob du pointeur virtuel que dirige l'utilisateur.

$$C_R(d) = \begin{cases} 1 & \text{Si } d < r' \\ f_R(d) & \text{Si } r' < d < r'' \\ 0 & \text{Si } d > r'' \end{cases} \quad (4.7)$$

$$C_T(d) = \begin{cases} 0 & \text{Si } d < r' \\ f_T(d) & \text{Si } r' < d < r'' \\ 1 & \text{Si } d > r'' \end{cases} \quad (4.8)$$

Les fonction f_R et f_T , définies sur $d \in [r' < d < r'']$, définissent la part de chaque transformation dans la zone de transition. Elles peuvent être linéaires ou non linéaires

et peuvent être choisies en fonction de la sensibilité désirée par l'utilisateur (à la manière des réglages de sensibilité d'une souris d'ordinateur). Vous pouvez voir un exemple dans la figure 75.

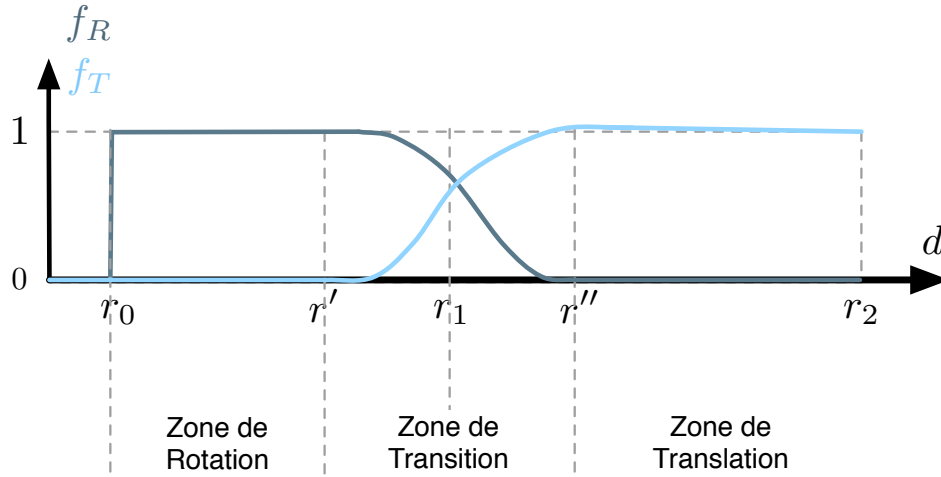


Figure 75: Exemple de variation des fonctions poids C_R et C_T .

Ainsi lorsque le pointeur est dans la zone de transition, l'utilisateur accomplit simultanément des translations et des rotations, mais pas dans les mêmes proportions, dépendant de quelle zone le pointeur est le plus proche. On remarquera que lorsque le pointeur est dans la zone de repos, les deux poids sont à zéros afin qu'il n'y ait aucune transformation d'appliquée sur le point de vue et par conséquent l'utilisateur sera à l'arrêt.

B - INFLUENCE DES FONCTIONS POIDS SUR LES TRANSFORMATIONS

A partir des modifications apportées par l'évolution du modèle, nous obtenons cette expression pour exprimer les translations sur le point de vue.

$$\vec{T} = I_T(d) \cdot \vec{u} = C_T(d) \cdot (d - r') \cdot \vec{u} \quad (4.9)$$

Avec :

$$\begin{aligned} I_T(d) &= C_T(d) \cdot (d - r') \\ d &= \|\vec{V}\| \\ 0 &\leq C_t(d) \leq 1 \end{aligned} \quad (4.10)$$

Où :

- $I_t(d)$ exprime la vitesse de translation,
- d est la distance séparant le pointeur de l'origine du blob,
- $C_T(d)$ est la fonction poids associée à la translation.

Afin de résoudre le second problème cité, nous avons également introduit une vitesse de rotation, ainsi, lors d'un changement rapide d'orientation, le point de vue est amené

progressivement à sa nouvelle orientation et par conséquent ce changement sera bien mieux perçu par l'utilisateur.

$$I_R(d) = C_R(d).d \quad (4.11)$$

C - CONCLUSION

Le résultat de cette évolution du modèle est une transition fluide entre les différentes transformations. Il n'y donc a priori aucune raison pour que la navigation soit mal vécue par l'utilisateur.

4.4.4 Fly-Over pour la tâche de sélection : Fly-Over-S

Lorsque la technique Fly-Over est utilisée pour la tâche de sélection, elle utilise une méthode hybride de pointage basée sur le *raycasting* et la *main virtuelle simple*.

Par défaut, la technique dite de la *main virtuelle simple* a été choisie comme méthode de sélection. Cette technique consiste à venir toucher, à l'aide d'un pointeur, un objet afin de le sélectionner. Si la main virtuelle simple ne renvoie aucune sélection, on utilise la technique du *raycasting* mais dans une version *2D picking*, c'est à dire que le rayon est toujours aligné par rapport à la direction dans laquelle regarde l'utilisateur.

REMARQUE 1 : On peut penser que ce type de technique de sélection est, d'un point de vue pratique, la technique *HOMER*, mais ce n'est pas exact. En effet, la technique *Homer* est une technique qui utilise le *raycasting* pour la sélection et qui utilise la *main virtuelle simple* pour la manipulation. Alors que dans notre cas, la main virtuelle simple peut aussi être utilisée pour la sélection. De plus le *raycasting* utilisé est la version 2D et non 3D.

REMARQUE 2 : L'utilisateur n'est pas perturbé par ce mode de fonctionnement car il est similaire au pointage à l'aide d'une souris, à la différence près qu'il peut aussi contrôler la position en profondeur de son pointeur.

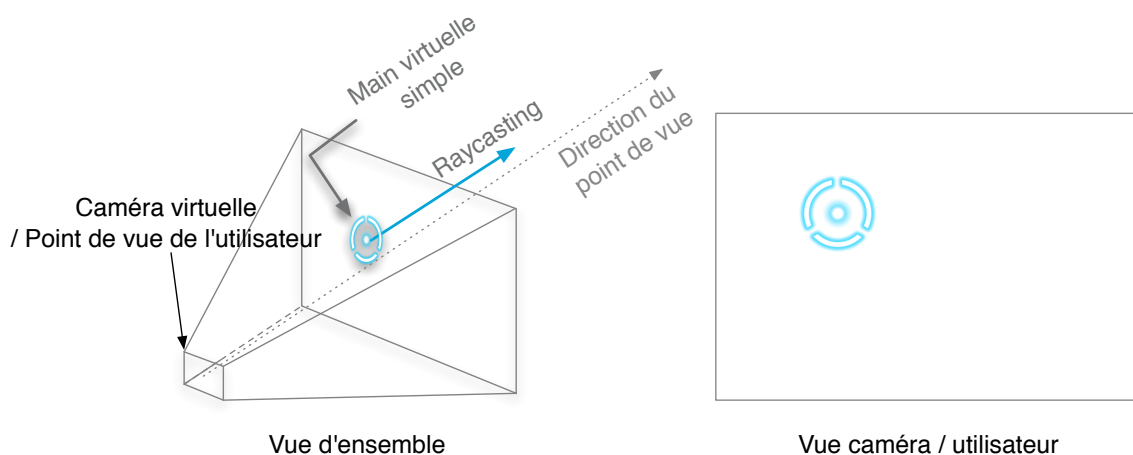


Figure 76: Disposition du pointeur lors de la selection.

L'avantage de pouvoir contrôler la profondeur de son pointeur, est que l'utilisateur peut aller positionner très aisément son pointeur sur un objet qui est en partie occulté par un autre (voir figure 77). Sans cette possibilité, l'utilisateur est obligé de passer en mode navigation afin de déplacer son point de vue et ainsi pouvoir avoir une meilleure vue sur l'objet qui était précédemment occulté, ou bien manipuler très précisément le pointeur afin de le placer sur l'objet en arrière-plan.

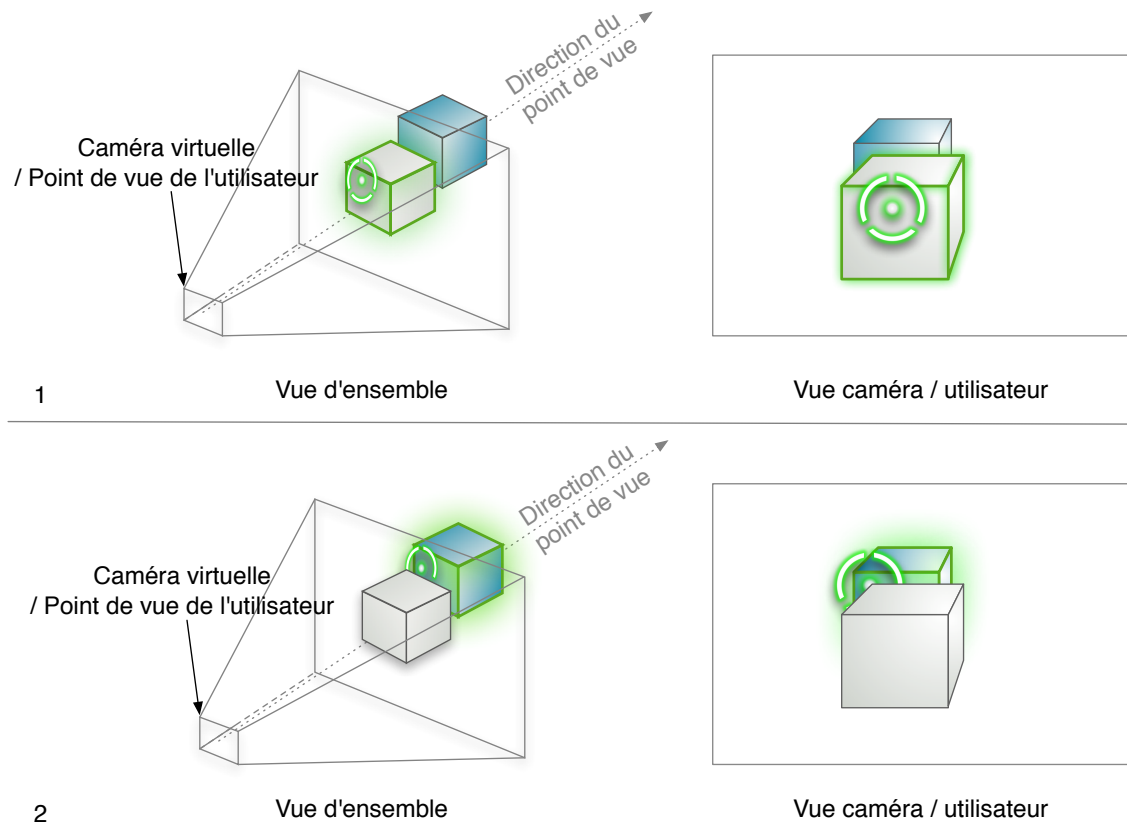


Figure 77: Exemple de sélection avec occultation - L'utilisateur peut sélectionner un objet qui est situé derrière un autre en jouant sur la profondeur du spot.

Fly-Over pour la tâche de sélection permet aussi de sélectionner plusieurs objets avant de les manipuler. Ainsi, on distingue deux phases lors d'une tâche de sélection :

1. Dans un premier temps, l'utilisateur sélectionne les objets qu'il souhaite manipuler par la suite. A chaque fois qu'il en sélectionne un, un halo de couleur verte apparaît autour de celui-ci afin de valider la sélection.
2. Dans un deuxième temps, l'utilisateur doit verrouiller sa sélection, c'est-à-dire l'ensemble des objets sélectionnés, afin d'indiquer au système qu'il est prêt à passer à la tâche de manipulation.

REMARQUE : Là encore, l'utilisateur n'est pas perturbé par ce mode de fonctionnement, car il est similaire à la gestion des fichiers dans un système d'exploitation. Il est, en effet, tout à fait possible de sélectionner un ensemble de fichiers et d'effectuer une action sur ceux-ci.

4.4.5 Fly-Over pour la tâche de manipulation : Fly-Over-M

Pour la version Fly-Over dédiée à la manipulation c'est la technique de la *main virtuelle simple* qui a été utilisée, du moins quand on manipule l'objet dans un espace assez confiné autour du blob et dans ce cas-là le point de vue reste fixe. Dès lors qu'on veut déplacer l'objet d'un endroit à un autre, on utilise Fly-Over pour la navigation à la différence que la sélection est attachée au pointeur et par conséquent le point de vue est mobile.

4.4.6 Fly-Over pour la tâche de contrôle d'application : Fly-Over-CA

D - GESTION DES ÉVÉNEMENTS AU SEIN DE FLY-OVER

Fly-Over intègre un gestionnaire d'événements basé sur la reconnaissance de gestes simples et évocateurs. Nous avons offert cette possibilité pour plusieurs raisons :

- Pallier le manque de boutons sur certains périphériques;
- Étendre les possibilités d'interaction en fonction des tâches de l'interaction 3D;
- Respecter la continuité de l'interaction 3D.

Nous avons défini dix événements correspondant à dix gestes répartis dans deux catégories les secousses et les balayages (respectivement *shake* et *sweep* en anglais).

Les secousses correspondent à des gestes consistants à secouer légèrement l'effecteur soit suivant un des trois axes, soit dans tous les sens. Cela constitue donc un total de quatre gestes de type secousse correspondant à quatre événements : *Shake*, *ShakeX*, *ShakeY* et *ShakeZ* (représentés dans la figure 78).

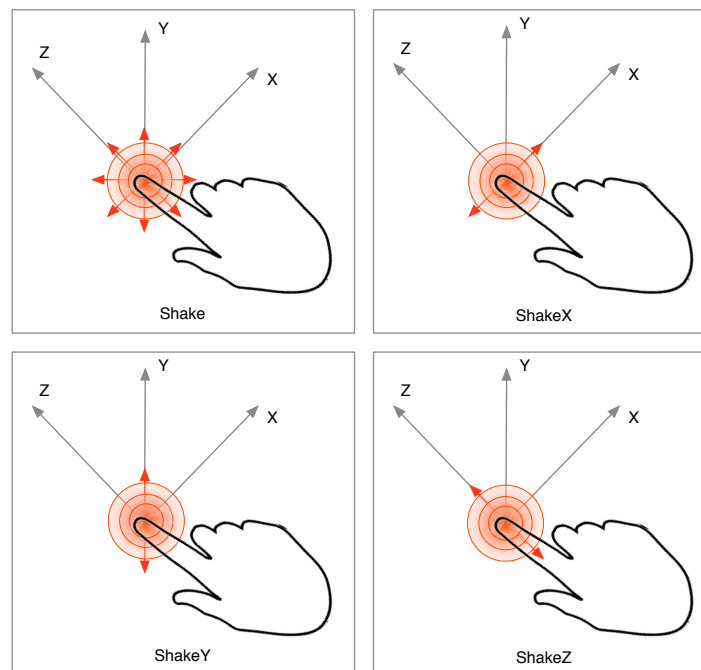


Figure 78: Aperçus des gestes de catégorie secousses.

Les balayages correspondent à des gestes consistant à faire glisser l'effecteur suivant un des trois axes, comme si l'on souhaitait balayer ou caresser quelque chose. On distingue aussi le sens du geste suivant l'axe ce qui nous amène à un total de six événements : *Sweep+X*, *Sweep+Y*, *Sweep+Z*, *Sweep-X*, *Sweep-Y* et *Sweep-Z* (représentés dans la figure 79).

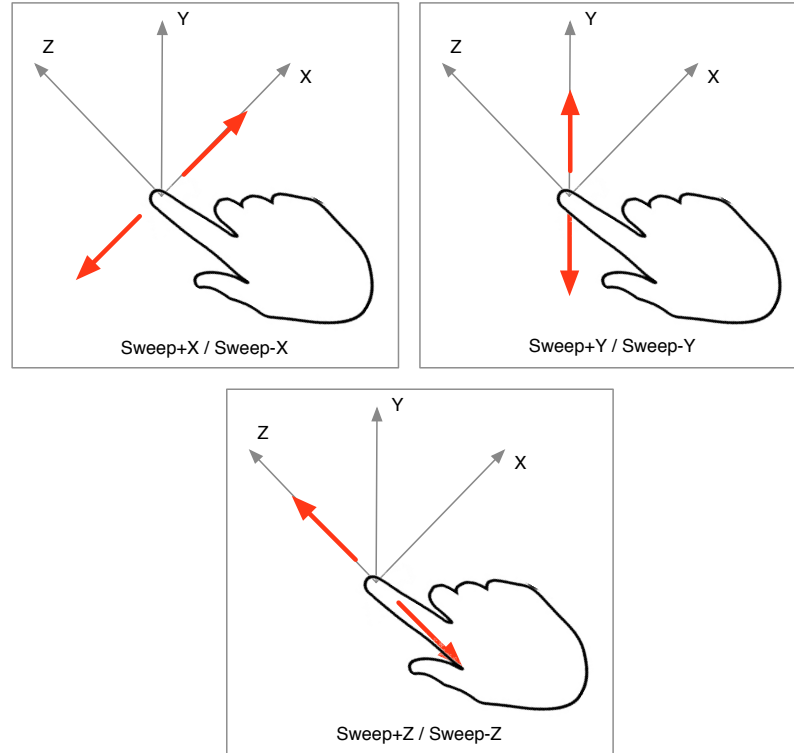


Figure 79: Aperçus des gestes de catégorie balayage.

4.4.6.1 Événements spécifiques à la tâche de navigation

- *Shake* : Initialise le repère de Fly-Over.
- *Sweep+Z* : Augmente le gain pour se déplacer.
- *Sweep-Z* : Diminue le gain pour se déplacer.
- *Sweep+X* : Effectue une rotation à 90° du point de vue suivant l'axe Y.
- *Sweep-X* : Effectue une rotation à -90° du point de vue suivant l'axe Y.
- *Sweep+Y* : Effectue une rotation à 90° du point de vue suivant l'axe X.
- *Sweep-Y* : Effectue une rotation à -90° du point de vue suivant l'axe X.

4.4.6.2 Événements spécifiques à la tâche de sélection

- *ShakeZ* : Sélectionne un objet.
- *Shake* : Verrouille la sélection.
- *Sweep+Z* : Effectue un zoom sur la sélection.
- *Sweep-Z* : Permet de s'éloigner de la sélection.
- *Sweep+X* : Oriente le point de vue vers le prochain objet sur la gauche.
- *Sweep-X* : Oriente le point de vue vers le prochain objet sur la droite.

- *Sweep+Y* : Oriente le point de vue vers le prochain objet vers le bas.
- *Sweep-Y* : Oriente le point de vue vers le prochain objet vers le haut.

4.4.6.3 *Événements spécifiques à la tâche de manipulation*

- *Shake* : Initialise le repère de Fly-Over.
- *Sweep+Z* : Augmente le gain pour se déplacer.
- *Sweep-Z* : Diminue le gain pour se déplacer.
- *Sweep+X* : Effectue une rotation à 90° du point de vue suivant l'axe Y.
- *Sweep-X* : Effectue une rotation à -90° du point de vue suivant l'axe Y.
- *Sweep+Y* : Effectue une rotation à 90° du point de vue suivant l'axe X.
- *Sweep-Y* : Effectue une rotation à -90° du point de vue suivant l'axe X.

4.4.7 *Discussion autour des tâches de Fly-Over*

On peut se poser la question de l'intérêt d'utiliser une approche anthropomorphique de l'interaction 3D dans Fly-Over. On entend par anthropomorphique, le fait que l'utilisateur doit dans la plupart des cas se déplacer jusqu'à l'objet pour le manipuler, comme il est habitué à le faire dans la réalité.

EXEMPLE: *Dans la réalité un être humain qui veut boire, va se diriger vers son placard, ouvrir la porte de celui-ci et saisir le verre pour enfin le manipuler.*

La raison de cette approche est simple, c'est qu'un utilisateur sera toujours moins perturbé en réalisant une tâche de manière que l'on qualifie d'habituelle, plutôt que de la réaliser par des méthodes que l'on qualifie de non naturelles. De plus, une telle approche est nécessaire dans les environnements de réalité mixte, dans lesquels le réalisme est obligatoire. Il est évident qu'une telle approche sera grandement appréciée de la part des utilisateurs novices. De la part, des utilisateurs experts, on s'attend à ce que ceux-ci ne soient pas perturbés, mais qu'ils se sentent éventuellement frustrés par la perte de temps que cette approche procure.

4.4.8 *Les assistances multimodales de Fly-Over pour l'utilisateur*

La technique d'interaction 3D Fly-Over propose un certain nombre d'assistances pour l'utilisateur ayant comme rôle principal d'aider celui-ci lorsqu'il interagit avec l'environnement virtuel. Ces assistances sont multimodales afin de communiquer de manière transparente et que quel que soit l'utilisateur. Cette communication se doit d'être exclusive et ne doit véhiculer aucun doute quant à l'interprétation par l'utilisateur des informations, que ces assistances sont censées transmettre. Nous avons défini deux sortes d'assistances, les assistances que nous qualifierons de directes et les indirectes.

4.4.8.1 Les assistances indirectes à l'utilisateur

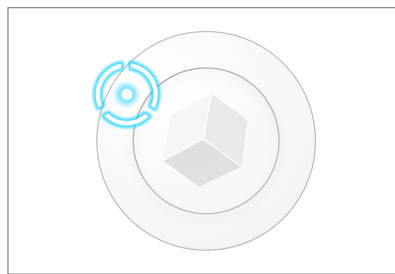
Les assistances indirectes, représentent les assistances au niveau sensori-moteur, c'est-à-dire sur les différentes modalités (vision, retour haptique, son, etc.). Ces assistances font généralement appel à des schèmes ou bien à l'inconscient collectif, afin qu'il n'y ait pas besoin d'explication sur le rendu de chacune ou qu'elles ne perturbent pas l'utilisateur.

A - LES ASSISTANCES SONORES

Lors du passage d'une tâche d'interaction 3D à une autre, le nom de la tâche est énoncé par une voix de synthèse, afin de signaler à l'utilisateur le changement de mode de Fly-Over. Des événements sonores ont lieu lors de la sélection d'un objet et lors de sa manipulation. Ainsi, un petit *clic* sonore retenti lors de la sélection et le bruit d'un cadenas que l'on ferme retenti lorsqu'on verrouille la sélection afin de passer en mode manipulation. Lors du déverrouillage de la sélection, dans un souci de logique, c'est un bruit de cadenas que l'on déverrouille qui est entendu.

B - LES ASSISTANCES VISUELLES

De nombreuses assistances visuelles sont présentes afin d'aider de manière plus ou moins appuyée l'utilisateur.



Vue caméra / utilisateur

Figure 80: Aperçu de la boussole par rapport à la position du pointeur.

Boussole: Lors de la navigation un cube apparaît telle une boussole afin de fournir à l'utilisateur un repère visuel. Le cube a été choisi car c'est un volume géométrique simple possédant des arêtes (contrairement à la sphère), l'utilisateur peut donc comprendre les transformations effectuées sur ce volume et ainsi avoir un meilleur sens de l'orientation dans l'espace.

États du spot: Le pointeur que dirige l'utilisateur avec sa main est un spot qui peut servir d'aide visuelle à l'utilisateur en utilisant différentes couleurs ou bien en pulsant. Ainsi le spot est de couleur :

- bleue, lorsqu'il n'y a aucun événement particulier;
- verte, lorsque un objet peut être sélectionné par l'utilisateur;
- orange, lorsqu'un événement est sur le point de se produire;
- rouge, lorsqu'on est en train de manipuler.



Figure 81: Les différents états du pointeur de Fly-Over en fonction du contexte.

On notera que ce code couleur est souvent utilisé dans ce sens : bleu rien à signaler, vert pour une invitation à agir, jaune pour avertir et rouge pour faire attention ou arrêter une action. Lorsque l'utilisateur positionne le spot sur un objet qui peut être manipulé, celui-ci se met à pulser, c'est-à-dire que son facteur d'échelle diminue puis augmente de manière régulière. Cette métaphore est généralement bien perçue par les utilisateurs car bon nombre d'appareils émettent des bruits ou font agiter des indicateurs en présence d'un objet (radar de recul d'une voiture, compteur Geiger). Un aperçu des différents états du pointeur peut être vu sur la figure 81.

Interface graphique: Des bandes noires situées en haut et en bas du point de vue l'utilisateur sont affichées afin de permettre une lecture aisée des informations sur l'interaction 3D quel que soit l'environnement représenté (figure 82).

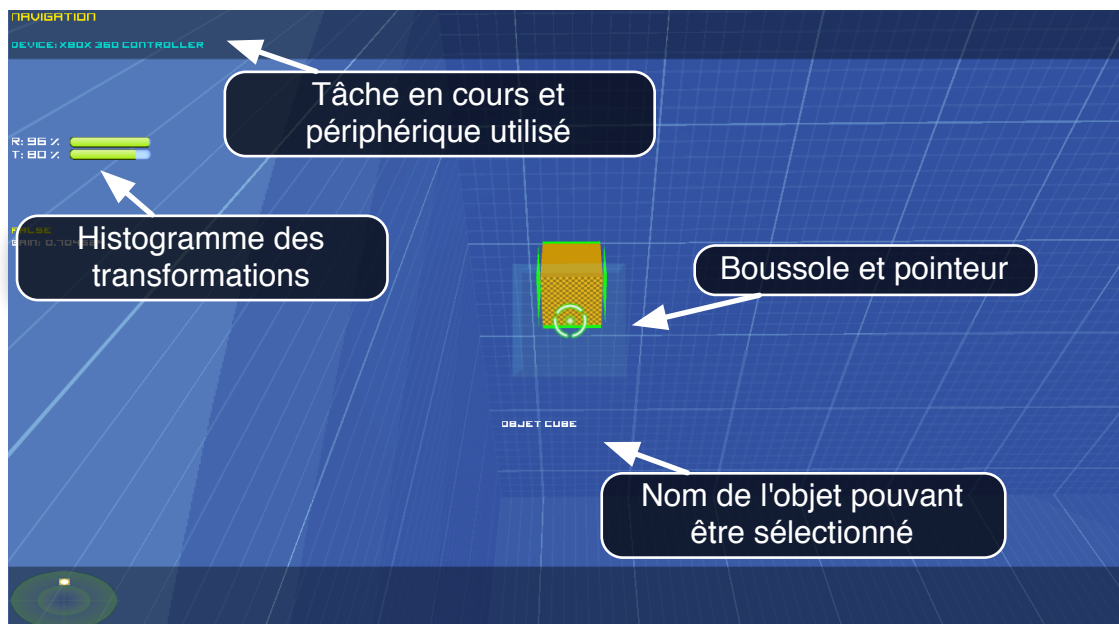


Figure 82: Interface graphique de Fly-Over.

Mais elles changent également de couleur durant l'interaction 3D (voir figure 83 et 84). Ainsi lorsque l'utilisateur verrouille la sélection, de la même manière que le spot devient rouge, les bandes deviennent également de couleur rouge afin de prévenir l'utilisateur qu'il est en train de réaliser une tâche de manipulation.

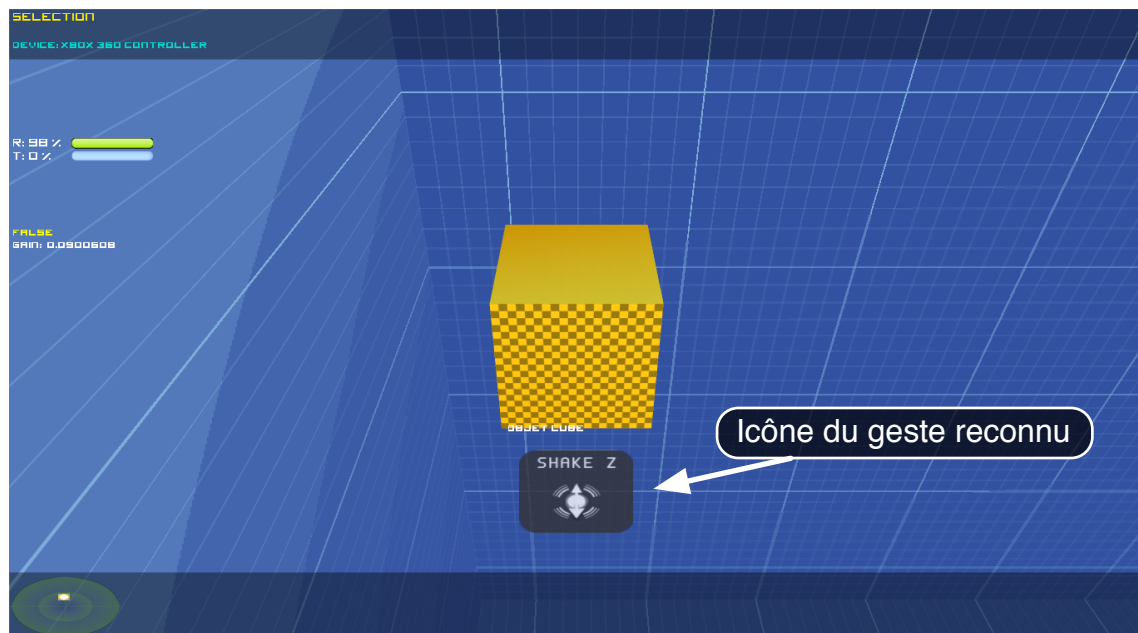


Figure 83: Interface graphique de Fly-Over lors de la tâche de sélection.

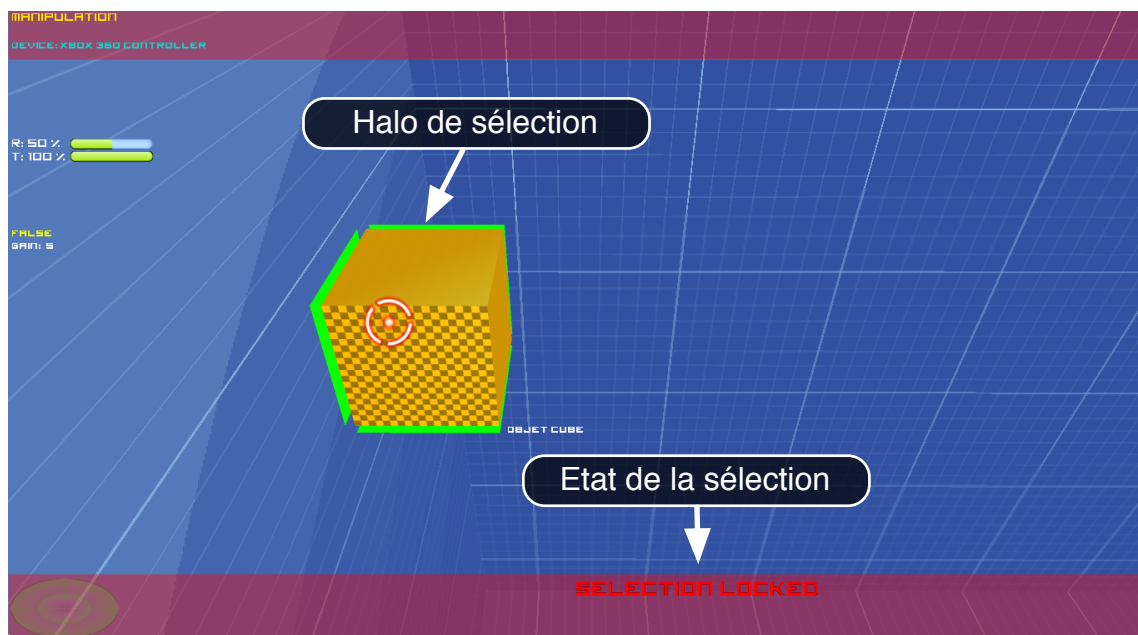


Figure 84: Interface graphique de Fly-Over lors de la tâche de manipulation.

L'interface graphique permet d'afficher différentes informations, telles que:

- Les histogrammes des transformations appliquées sur la caméra virtuelle (rotations et translations).
Ces histogrammes permettent d'éventuellement aider l'utilisateur à comprendre comment la technique d'interaction 3D Fly-Over se comporte en tâche de navigation lors de la phase d'apprentissage.
- Le nom du périphérique utilisé.
Cette information est surtout présente pour vérifier le bon fonctionnement du système.

- Le nom de la tâche de l'interaction 3D en cours.
Là encore cette information est présente plus pour vérifier le bon fonctionnement du système plutôt que pour venir en aide à l'utilisateur.
- L'état de la sélection (Sélection verrouillée ou non).
Information essentielle, pour que l'utilisateur comprenne qu'il peut sélectionner ou bien manipuler des objets dans l'environnement virtuel.
- Le nom de l'objet pouvant être saisi.
Cette information n'est pas essentielle, cependant elle permet d'éventuellement lever l'ambiguïté sur l'objet qui est saisi dans le cas où plusieurs objets sont proches.
- Des icônes indiquant le geste reconnu par le système.
Le simple fait d'afficher une icône témoignant du geste reconnu par le système est important pour l'utilisateur car il permet à celui-ci d'être assuré que le geste qu'il a réalisé et bien celui perçu par le système.

C - LES ASSISTANCES HAPTIQUES

Lorsqu'un périphérique à retour d'effort est utilisé, le retour haptique est activé afin de détecter les collisions entre le pointeur que manipule l'utilisateur avec l'environnement virtuel. Le contact avec un obstacle produit une force de type choc élastique. C'est-à-dire que lors d'un choc la force de retour est proportionnelle à la force de contact. Plus on se rapprochera de l'obstacle et plus la force appliquée sera importante. Cependant pour des raisons de sécurité et protection du périphérique haptique on limite la force maximale appliquée à 10 Newtons. Le retour haptique est aussi en partie utilisé pour replacer l'effecteur que manipule l'utilisateur vers le centre du blob. Ainsi, l'utilisateur à l'impression d'utiliser une souris 3D ou bien un joystick qui reviennent toujours à leur position du repos. Cette force appliquée n'est pas très importante, elle est juste suffisante pour ramener l'effecteur vers sa position d'origine.

4.4.8.2 Les assistances directes à l'interaction 3D

Fly-Over possède ce que l'on appelle des assistances *direct* à l'interaction 3D. Le terme *directes* est présent ici pour surligner l'immédiateté de l'agissement de ces assistances sur l'interaction 3D car elles influent au plus bas de niveau de compréhension du système, c'est-à-dire sur les données. A la différence des précédentes assistances qui, elles, agissent dans un domaine spécifique, qui ensuite, influe sur l'interaction 3D (exemple le retour haptique qui exerce des contraintes sur la main de l'utilisateur, et la main qui finalement agit sur l'interaction 3D).

A - GAIN ADAPTATIF

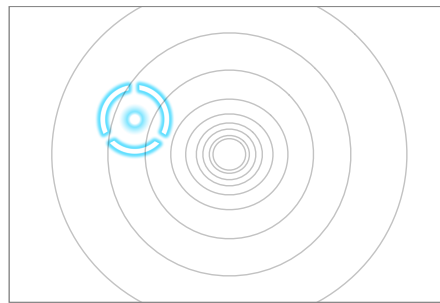
Afin d'aider l'utilisateur de la manière la plus transparent possible, il est intéressant d'utiliser ce qu'on appelle un gain adaptatif. Un gain est une valeur avec laquelle, généralement, on multiplie un paramètre afin de le pondérer. Il peut être intéressant de faire varier un gain en fonction des situations afin de faciliter les choses pour l'utilisateur.

Ainsi, nous avons défini un gain adaptatif sur la position du pointeur de Fly-Over que manipule l'utilisateur, en fonction de la tâche de l'interaction 3D et la position du pointeur dans l'espace. Le tableau 1 récapitule les différentes valeurs du gain adaptatif appliquées à la position du pointeur que l'utilisateur manipule en fonction.

REMARQUE : Ces gains ont été fixés empiriquement à partir d'observation de plusieurs profils de personnes utilisant la technique d'interaction 3D Fly-Over.

	GAIN MAX	AXES	FONCTION GAIN
NAVIGATION	0.75	x,y,z	$f(x) = 1x$
SÉLECTION	3.0	x	$f(x) = \frac{3}{2}(1 - \cos \frac{5x}{\pi})$
	3.0	y	$f(x) = \frac{3}{2}(1 - \cos \frac{5x}{\pi})$
	1.0	z	$f(x) = 1x$
MANIPULATION	3.0	x,y,z	$f(x) = 3x$

Table 1: Résumé des différents gains en fonction de la tâche de l'interaction 3D et de la position du pointeur.



Vue caméra / utilisateur

Figure 85: Variation du gain dans l'espace : Plus la différence de rayon entre deux cercles est grande, plus le gain est grand.

On peut remarquer que lors de la tâche de navigation, le gain est constant afin de ne pas privilégier un sens de navigation, car nous voulons que l'utilisateur puisse se déplacer aisément dans n'importe quelle direction. En revanche, lors de la tâche de sélection, le gain sur l'axe z est privilégié par rapport aux autres axes, tout du moins, lorsque la composante x de la position du pointeur est située vers le centre du repère de Fly-Over ($x \in [-1; 1]$). Nous avons réglé le gain adaptatif de cette manière-là afin que l'utilisateur puisse sélectionner plus aisément les objets qui sont en face de lui, ainsi que pour favoriser les gestes : *Sweep+Z*, *Sweep-Z* et *ShakeZ*. En contrepartie, dès lors que l'on déplace le pointeur plus vers l'extérieur du centre du repère, les composantes x et y seront favorisées par le biais d'un gain plus important par rapport à la composante z. Enfin, lors de la tâche de manipulation, tout comme pour la tâche de navigation, le gain est constant sur les trois composantes, afin de ne favoriser aucun des axes lors de la manipulation.

B - ASSISTANCES AUX GESTES

Des guides virtuels sont présents afin de contenir le mouvement du pointeur suivant certains axes lorsque le système détecte que le mouvement réalisé par la main de l'utilisateur tente de suivre un axe et donc en l'occurrence essaie de réaliser un des gestes précédemment indiqués. La raison de la présence de ces guides est d'apporter une cohérence entre ce que souhaite exécuter l'utilisateur et le mouvement du pointeur à l'écran. Par exemple quelqu'un qui exécute un geste de type *balayage vers la droite* s'attend à ce que le pointeur suive l'axe des X.

L'estimation d'un mouvement suivant un des 3 axes (x, y, ou z) peut être déterminée en regardant la composante majoritaire du vecteur accélération \vec{a} entre 2 instants t_1 et t_2 avec $t_2 - t_1 > 500$ millisecondes.

$$\vec{a} = [a_x \ a_y \ a_z]^T \quad (4.12)$$

En normalisant le vecteur \vec{a} on obtient le vecteur accélération $\vec{a_N}$ dont les composantes exprimées en pourcentage reflètent l'influence de chaque composante sur le vecteur.

$$\vec{a_N} = \frac{\vec{a}}{\|\vec{a}\|} = [a_{Nx} \ a_{Ny} \ a_{Nz}]^T \quad (4.13)$$

La normalisation nous permet d'estimer la part de chaque composante indépendamment de l'ordre de grandeur du vecteur accélération. Si une composante est supérieure à ε ou inférieure à $-\varepsilon$ alors on considère que le mouvement doit être limité à suivre l'axe correspondant.

On a donc un mouvement plutôt sur l'axe :

X si $\max(\vec{a_N}) = a_{Nx}$ et $|a_{Nx}| > \varepsilon$,

Y si $\max(\vec{a_N}) = a_{Ny}$ et $|a_{Ny}| > \varepsilon$,

Z si $\max(\vec{a_N}) = a_{Nz}$ et $|a_{Nz}| > \varepsilon$.

On exerce une contrainte sur le mouvement du pointeur en additionnant la position précédente à la delta position multipliée par une matrice de 0 avec juste un 1 sur la composante que l'on souhaite garder.

Ainsi :

Si mouvement suivant l'axe X :

$$\vec{P_i} = \vec{P_{i-1}} + \vec{\Delta_i} \cdot [1 \ 0 \ 0]$$

Si mouvement suivant l'axe Y :

$$\vec{P_i} = \vec{P_{i-1}} + \vec{\Delta_i} \cdot [0 \ 1 \ 0]$$

Si mouvement suivant l'axe Z :

$$\vec{P_i} = \vec{P_{i-1}} + \vec{\Delta_i} \cdot [0 \ 0 \ 1]$$

Avec :

$\vec{P_i}$: Le vecteur position courante du pointeur.

$\vec{P_{i-1}}$: Le vecteur position précédente du pointeur.

$\vec{\Delta_i}$: La delta-position entre la position courante et la position précédente du pointeur, soit : $\vec{P_i} - \vec{P_{i-1}}$.

4.5 L'UNITÉ D'INTERACTION 3D ADAPTATIVE

4.5.1 Introduction

Nous avons d'un côté l'UCCD qui nous permet de gérer les données et d'assurer la continuité des données et de l'autre nous avons une technique d'interaction 3D homogène qui permet de réaliser toutes les tâches de l'interaction 3D. Cependant, ces deux seuls composants ne suffisent pas pour assurer l'objectif principal qui est la continuité de l'interaction 3D. En effet, le passage d'une tâche à l'autre n'est pas automatique et c'est à l'utilisateur d'en faire la requête. Le problème est justement que l'utilisateur doit en faire la requête. Plusieurs solutions existent pour répondre à ce problème, comme par exemple la commande vocale, l'utilisation de widgets ou bien tout simplement l'appui sur des boutons. Cependant ces solutions coupent l'utilisateur dans son élan et par conséquent empêche la continuité de l'interaction 3D. Ce que nous proposons pour préserver le continuum de l'interaction 3D est de concevoir un composant capable d'estimer la tâche de l'interaction 3D que souhaite réaliser l'utilisateur. Pour cela, nous utilisons d'une part un observateur de contexte, qui surveillera certains paramètres et, d'autre part, un système décisionnel capable, à partir des paramètres observés, de prendre une décision quant à la tâche de l'interaction 3D que souhaite réaliser l'utilisateur.

4.5.2 Paramètres observables

Un observateur de contexte est un élément du système dont le rôle est de surveiller l'état de différents paramètres afin de les communiquer au système de décision. Dans notre cas, il observe un ensemble de paramètres relatifs au point de vue de la caméra virtuelle, la présence du pointeur contrôlé par l'utilisateur dans le blob d'interaction 3D de Fly-Over, ainsi qu'à l'état de la sélection qui ont été choisis afin de permettre au système de décision de choisir ou non le passage d'une tâche à une autre. Vous pouvez observer une représentation visuelle des paramètres dans le contexte de la technique Fly-Over sur la figure 86

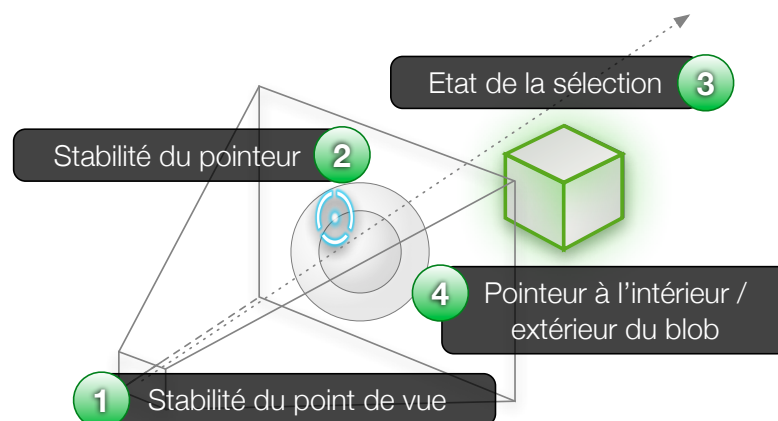


Figure 86: Les 4 paramètres utilisés par l'unité d'interaction 3D adaptative.

1. *Stabilité du point de vue :*

La stabilité du point de vue est un paramètre de type booléen, permettant de déterminer si la caméra virtuelle (représentant le point de vue de l'utilisateur dans l'environnement virtuel) est en mouvement ou non.

2. *Stabilité du pointeur :*

La stabilité du pointeur est en tout point semblable au paramètre précédent à la différence près qu'il reflète la présence d'un mouvement du pointeur virtuel et non de la caméra virtuelle.

3. *Etat de la sélection :*

Ce paramètre renseigne tout simplement si la sélection est : *verrouillée* (prête à être manipulée) ou *déverrouillée* (prête à être relâchée).

4. *Pointeur à l'intérieur/extérieur du blob d'interaction 3D :*

Ce paramètre nous informe sur la présence ou non du pointeur contrôlé par l'utilisateur au sein du blob d'interaction 3D de la technique Fly-Over.

4.5.3 Système de décision

Le système de décision a pour principale fonction de gérer le passage d'une tâche de l'interaction 3D à une autre. Pour cela il s'appuie sur un ensemble de règles établissant une table de décision. Le modèle de Fly-Over nous a permis de simplifier grandement cet aspect, car le découpage en sous-espaces permet d'évaluer rapidement quelle tâche veut réaliser l'utilisateur à partir de la position du pointeur dans les différents sous-espaces. La tâche par défaut est celle de navigation, l'utilisateur réalise des mouvements avec sa main pour passer d'une zone à l'autre, effectuant ainsi des rotations et des translations à la caméra virtuelle.

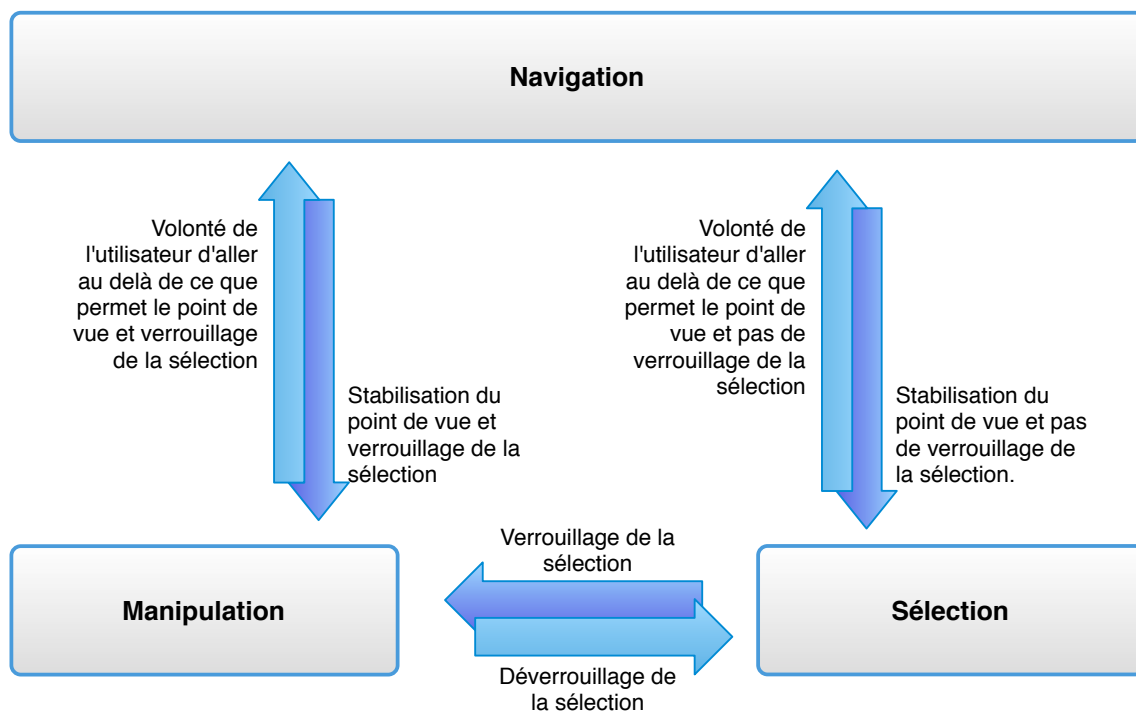


Figure 87: Règles pour basculer entre les tâches de l'interaction 3D.

La figure 87 et le tableau 2 expliquent les règles de décision permettant de passer d'une tâche de l'interaction 3D à une autre. Lorsque l'utilisateur commence à ralentir son déplacement au sein de l'environnement virtuel, en revenant vers l'origine du repère de Fly-Over, stabilisant ainsi son point de vue, le système bascule à la tâche de sélection. Le gain adaptatif précédemment cité est alors modifié de manière à ce que les déplacements du pointeur autour du centre du plan ($x; y$) soient moins importants qu'à la périphérie de celui-ci. L'utilisateur peut ainsi sélectionner les objets qu'il souhaite manipuler. Une fois cette phase effectuée, l'utilisateur peut verrouiller la sélection via la réalisation d'un geste. Le système bascule donc vers la tâche de manipulation. L'objet ou les objets sélectionnés sont attachés au graphe de scène du pointeur de Fly-Over et l'utilisateur peut donc les manipuler directement avec sa main. Si l'utilisateur souhaite déplacer le ou les objets vers un autre emplacement, il a tendance à vouloir placer sa main dans l'espace de manière à ce que le pointeur aille au-delà de ce que permet le point de vue. Le système reconnaît cette volonté et passe donc à la tâche de navigation, mais cette fois-ci l'objet est toujours attaché au pointeur. Lorsque l'utilisateur stabilise son point de vue, le système bascule de nouveau vers la tâche de manipulation afin qu'il puisse déposer le ou les objets à l'endroit où il le souhaite, de manière précise. Lorsqu'il déverrouille le ou les objets sélectionnés, le système repasse en mode de sélection.

DE LA TÂCHE A	À LA TÂCHE B	ÉTAT DE LA SÉLECTION	STABILITÉ PT. DE VUE	STABILITÉ DU POINTEUR	POINTEUR À L' INTÉRIEUR/EXTÉRIEUR DU BLOB
N	S	Déverr.	Oui	Oui	Intérieur
S	N	Déverr.	Oui	Non	Extérieur
S	M	Verr.	-	-	Intérieur
M	S	Déverr.	-	-	Intérieur
M	NM	Verr.	Oui	Non	Extérieur
NM	M	Verr.	Oui	Oui	Extérieur

Table 2: Table de décision pour passer d'une tâche de l'interaction 3D à une autre.

Avec :

N : Navigation

S : Sélection

M : Manipulation

NM : Navigation avec un objet (Manipulation)

- : Paramètre non pris en compte

4.6 L'UNITÉ DE CORRECTION ET COMBINAISON DES DONNÉES: UCCD

L'unité de correction et combinaison des données (UCCD) est un composant essentiel de notre système. En effet, il permet d'appliquer des traitements sur les données provenant des différents périphériques. Les résultats de ses traitements seront mis à

disposition sur le réseau sous la forme de nouveaux périphériques virtuels. Ce composant agit comme une couche d'abstraction des données et permet de représenter les différentes données possibles sous une forme standardisée. De plus, il offre une solution performante pour les utilisateurs non-experts désirant appliquer de manière simple, certains traitements sur des données, à l'aide d'un fichier de configuration XML répertoriant les différents traitements à effectuer.

4.6.1 Contexte dans lequel se positionne l'UCCD

Il existe deux grandes familles d'architecture de systèmes multi-capteurs en réalité virtuelle. Le premier type d'architecture (voir figure 88) est le plus courant, il consiste à connecter les différents périphériques directement au sein du logiciel de réalité virtuelle. La plupart du temps, cette solution est choisie car c'est la manière la plus simple et rapide de procéder. Malheureusement avec la multiplication des périphériques une telle architecture devient de plus en plus complexe et erratique. De plus cela nécessite de faire exécuter des tâches supplémentaires au logiciel de réalité virtuelle, ce qui peut l'amener à manquer de ressources pour interagir avec l'environnement virtuel dans des performances convenables. Cette architecture devient donc très rapidement inutilisable.

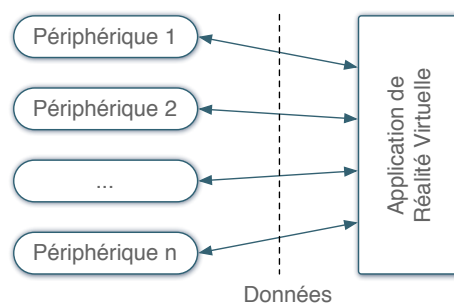


Figure 88: Architecture classique de système multi-capteurs en réalité virtuelle.

L'autre grande famille d'architecture de système multi-capteurs (voir figure 89) possède un composant qui sert d'intermédiaire entre les périphériques et le logiciel de réalité virtuelle. Il agit comme une couche d'abstraction des données, dans laquelle les données vont être regroupées par type et représentées sous une forme homogène.

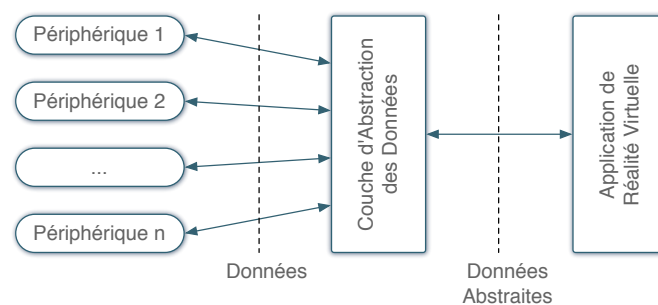


Figure 89: Architecture de système multi-capteurs en réalité virtuelle avec couche d'abstraction.

Cependant cette architecture souffre d'un problème majeur : il est impossible d'effectuer le moindre traitement sur les données à ce niveau-là et donc au final les traitements à appliquer se font au sein du logiciel de réalité virtuelle ce qui au final peut amener aux mêmes problèmes que dans le premier type d'architecture. Ce que nous proposons est de rajouter une partie traitement de données à la couche d'abstraction afin de décharger le logiciel de réalité virtuelle (voir figure 90) et ainsi centraliser les traitements sur les différents périphériques dans un seul et même composant.

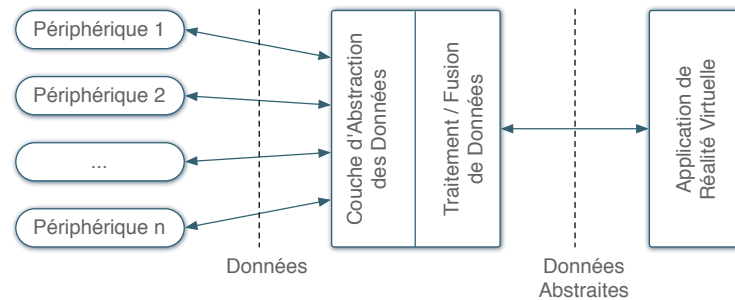


Figure 90: Architecture de système multi-capteurs en réalité virtuelle avec UCCD.

4.6.2 L'UCCD dans un système de réalité virtuelle

C - LES PÉRIPHÉRIQUES

L'utilisateur interagit avec l'environnement virtuel grâce à différents périphériques, capteurs ou interfaces qui communiquent tous à l'ordinateur par bus série (USB), interface spécifique (carte d'acquisition) ou bien par le réseau.

D - LE LOGICIEL DE RÉALITÉ VIRTUELLE

Le logiciel de réalité virtuelle correspond au programme chargé de représenter l'environnement virtuel sur un périphérique vidéo (casque, écran, projecteur) ainsi que de gérer l'interaction 3D c'est à dire de transformer l'interaction 3D matérielle, entre l'homme et les périphériques, en interaction 3D virtuelle, entre l'homme et l'environnement virtuel.

E - L'UNITÉ DE CORRECTION ET COMBINAISON DES DONNÉES

L'UCCD se place entre le logiciel de réalité virtuelle et les périphériques. Il est lui-même composé de 3 entités : une *application web* qui permet de générer un *fichier de configuration XML* qui sert au cœur du composant : l'*UCCD Core*.

4.6.3 Les différents composants de l'UCCD

4.6.3.1 Application Web

Nous utilisons une application web afin de fournir une interface graphique à l'utilisateur, pour qu'il puisse configurer l'UCCD. Au travers de formulaires web (case à cocher, liste déroulante, champ de saisie) l'utilisateur choisit les périphériques qu'il souhaite

utiliser, les traitements à appliquer et sur quels périphériques les appliquer. A l'issue de cette configuration, un fichier de configuration, de type XML, est généré.

4.6.3.2 Fichier de configuration

Toute la configuration de l'UCCD est décrite dans ce fichier XML. Ce fichier est composé de trois parties : initialisation des périphériques, description des périphériques virtuels et description des événements.

A - INITIALISATION DES PÉRIPHÉRIQUES

Cette partie du fichier de configuration (figure 91) est celle où sont décrits les différents périphériques à écouter et quelles données seront extraites.

```
<initialization>
  <name>
    EVR@ Default Configuration
  </name>
  <description>
    Default configuration for our VR platform
  </description>
  <usedDevices>
    <vrpn>
      <device class="tracker" type="VRPN" name="ART-Hand" />
      <device class="button" type="VRPN" name="ART-Flystick" />
      <device class="button" type="VRPN" name="Data Gloves" />
    </vrpn>
    <others>
      <device class="tracker" type="other" name="SPIDAR" />
      <device class="tracker" type="other" name="Wiimote" />
    </others>
  </usedDevices>
</initialization>
```

Figure 91: Description de l'initialisation des périphériques.

B - DESCRIPTION DES PÉRIPHÉRIQUES VIRTUELS

Dans cette partie (figure 92) sont décrits les périphériques virtuels, qui sont les résultats de traitements sur un ou plusieurs périphériques (qu'ils soient virtuels ou bien réels). Cette description contient donc la méthode mathématique à appliquer.

```
<devices>
  <device class="tracker" type="virtual" name="Hybrid Tracking">
    <method name="correspondance">
      <device name="ART-Hand" />
      <device name="SPIDAR" />
    </method>
  </device>
</devices>
```

Figure 92: Description des périphériques virtuels.

C - DESCRIPTION DES ÉVÉNEMENTS

Dans cette partie du fichier de configuration (voir figure 93) sont listés tous les événements que l'utilisateur veut détecter au sein de l'application de réalité virtuelle. Il suffit d'indiquer quelle(s) donnée(s) écouter et quelle(s) condition(s) les données doivent satisfaire pour déclencher l'événement.

```
<events>
  <event type="virtual" id="0" name="Closed fist" method="test">
    <device class="analog" type="VRPN" name="Data Gloves">
      <value>0</value>
    </device>
    <device class="button" type="VRPN" name="Flystick Button">
      <value>0</value>
    </device>
  </event>
</events>
```

Figure 93: Description des événements virtuels.

4.6.3.3 L'UCCD Core

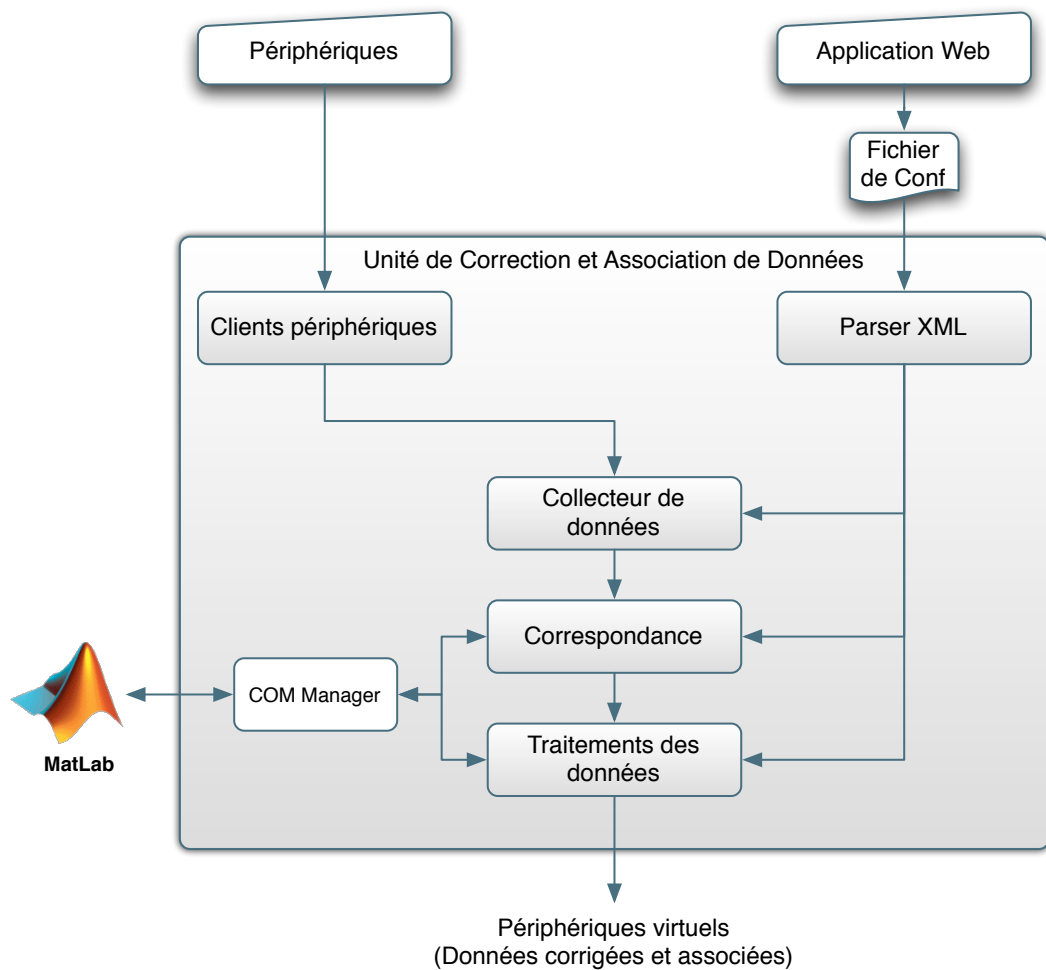


Figure 94: L'unité de correction et combinaison des données.

L'UCCD Core, représenté sur la figure 94 est le noyau de l'UCCD. A partir de l'interprétation du fichier de configuration XML, ce composant initialise les communications avec les différents périphériques. Une fois cette étape réalisée il appliquera les méthodes mathématiques à appliquer sur les données. Enfin il mettra à disposition les données calculées sur le réseau. D'un point de vue haut niveau, ce composant agit comme un périphérique virtuel ou un ensemble de périphériques virtuels avec lesquels l'application de réalité virtuelle communique. Il a été conçu comme un serveur TCP/IP multi-processus et multi-utilisateurs afin qu'il puisse être utilisé par tous types d'applications et tous types de plates-formes et ce, simultanément. De plus, une telle conception nous assure que toute la partie gestion matérielle et traitements sur les données, qui généralement consomment beaucoup de ressources, ne viendra pas influencer les performances de l'application de réalité virtuelle.

A - GESTIONNAIRE DE PÉRIPHÉRIQUES

Ce sous-composant permet de communiquer avec les différents périphériques par le biais de différents moyens : un client VRPN, l'utilisation d'APIs dédiées ou bien d'autres clients plus appropriés. Il représente la première couche d'abstraction.

B - LE COLLECTEUR DE DONNÉES

Le collecteur de données est le composant qui filtre les données provenant du gestionnaire de périphériques afin de ne laisser passer que les données à traiter, assurant ainsi de ne pas utiliser de ressources inutiles. Il représente la seconde couche d'abstraction.

C - CORRESPONDANCE

Cette partie s'occupe de ramener les données, provenant des différents périphériques et exprimées dans des référentiels différents, dans un nouvel espace de représentation commun à toutes. Pour cela il s'appuie sur l'ensemble des fonctions mathématiques dont le logiciel MATLAB® ([MathWorks](#)) dispose.

D - TRAITEMENTS DES DONNÉES

Dans ce composant sont appliqués les traitements décrits dans le fichier de configuration, sur les données. Là encore, on peut utiliser l'ensemble de la bibliothèque mathématique de MATLAB® ([MathWorks](#)).

E - LE CHARGEUR XML

Ce composant ouvre le fichier de configuration XML et vérifie son intégrité ainsi que le respect des standard W3C à l'égard des fichiers XML. S'il ne peut valider son intégrité, ce composant avertit l'utilisateur de la non validité du fichier de configuration et empêche l'UCCD de fonctionner afin d'éviter tout problème dû à une mauvaise interprétation du fichier.

F - LE PARSEUR XML

Ce composant découpe le fichier de configuration en trois parties présentées précédemment (initialisation des périphériques, description des périphériques virtuels et description des événements) et les transforme en processus parallèles qui s'appliqueront sur les données.

G - MANAGER COM ET MATLAB®

Afin de proposer le plus grand nombre de méthodes mathématiques à l'utilisateur de l'UCCD, nous y avons intégré un manager COM, dont le but de faire le pont entre l'UCCD et le logiciel mathématique MATLAB® (MathWorks). Ainsi, l'UCCD a accès à l'ensemble des capacités de calcul de MATLAB® (MathWorks).

4.6.4 Algorithme de traitement

La figure 95 représente un algorithme simplifié représentant les différentes étapes du processus de traitement des données dans l'UCCD. On peut apercevoir la boucle de traitement (étapes 4, 5 et 6) qui consiste à :

- Dans un premier temps, récupérer les données des différents périphériques,
- Puis, appliquer les traitements sur les données et les agglomérer sous la forme de périphériques virtuels,
- Enfin, mettre à disposition sur le réseau les données calculées provenant des différents périphériques virtuels.

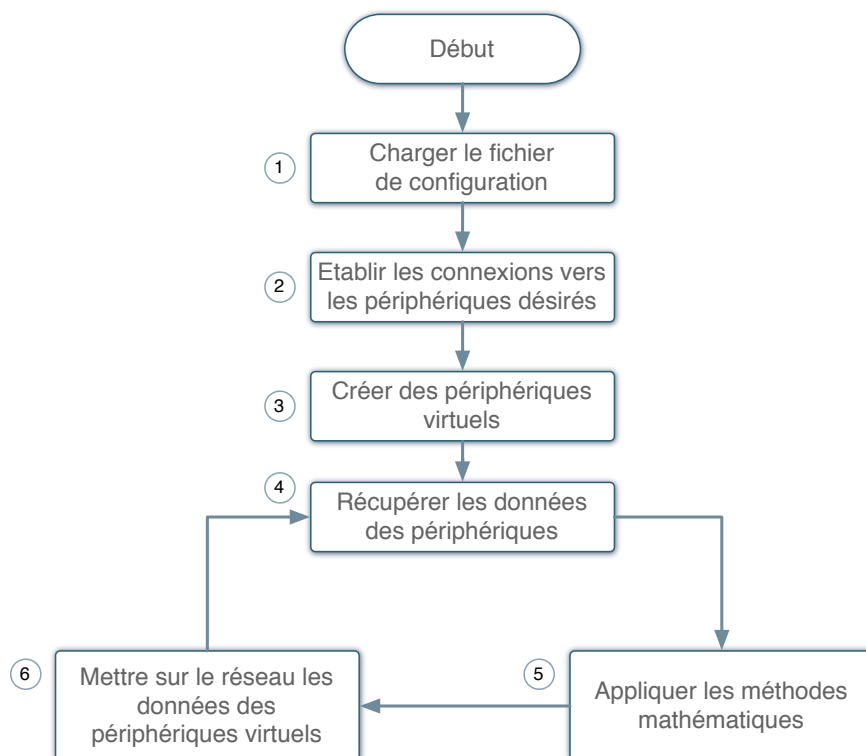


Figure 95: Algorithme de traitement simplifié.

4.6.5 Relations entre l'UCCD, l'UI3D-A et Fly-Over

L'unité de correction et combinaison de données permet de créer des périphériques virtuels dans lesquels les données peuvent disposer d'un ou plusieurs traitements. Ses sorties sont de types : vecteur, booléen, entier, réel ou bien chaîne de caractères. La technique d'interaction 3D Fly-Over nécessite seulement un vecteur position qui soit continu. C'est pour cette raison que le traitement en amont de Fly-Over doit être efficace. En utilisant l'UCCD pour concevoir ce vecteur position continu, on garantit le bon fonctionnement de Fly-Over.

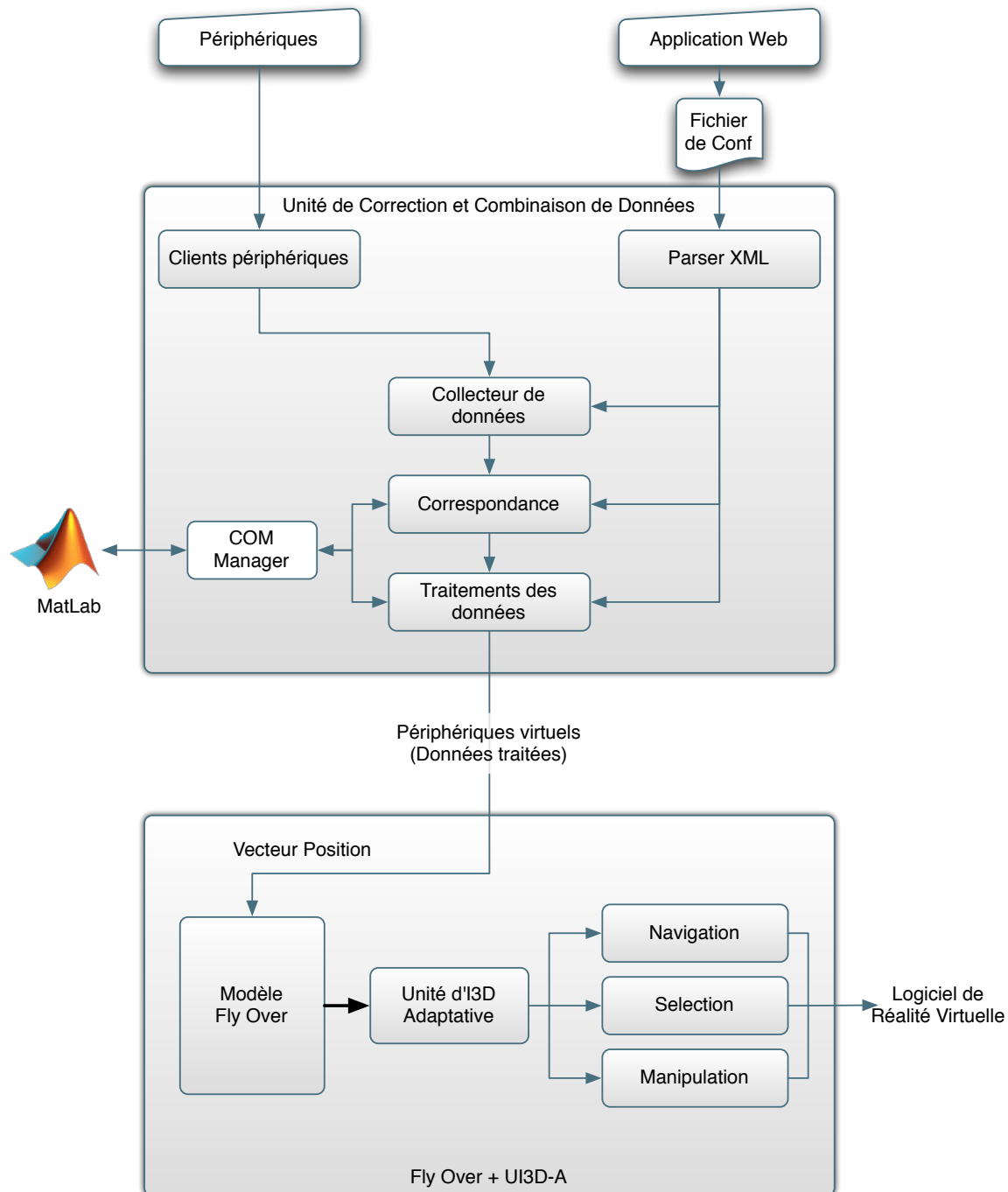


Figure 96: Interaction entre le l'UCCD et Fly-Over.

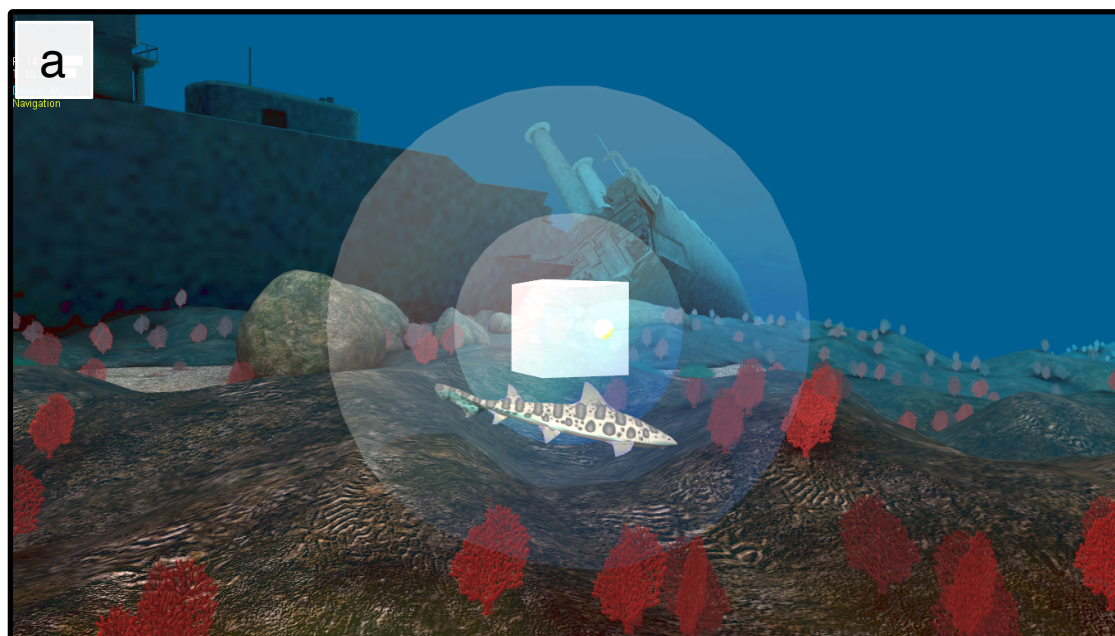
La figure 96 représente la manière dont s'architecturent ces différents composants. Cependant avoir un vecteur position continu n'est pas suffisant pour garantir une interaction 3D continue, dans la mesure où le basculement d'une tâche de l'interaction 3D à une autre doit se faire automatiquement. Comme nous l'avons vu dans la partie 4.5, c'est l'unité d'interaction 3D adaptative qui tient ce rôle. Il se situe entre le modèle générique de Fly-Over et les différents modules de Fly-Over permettant de gérer les trois tâches de l'interaction 3D : la navigation, la sélection et la manipulation.

4.7 BILAN

4.7.1 Notre système dans différentes applications de réalité virtuelle

La figure 97 présente différentes applications de réalité virtuelle utilisant notre système d'interaction 3D adaptative.

- La figure 97.a démontre l'utilisation de notre système dans un environnement sous-marin dédié à la collecte de médias (sons, vidéos, photos) et leur positionnement sur l'environnement 3D.
- La figure 97.b montre la technique Fly-Over dans un environnement fermé, représentant notre laboratoire et nécessitant une navigation précise.
- Nous avons pu également tester notre système avec une application permettant de créer une représentation 3D crédible d'une molécule d'ADN (figure 97.c), respectant des contraintes inhérentes à la biologie moléculaire.



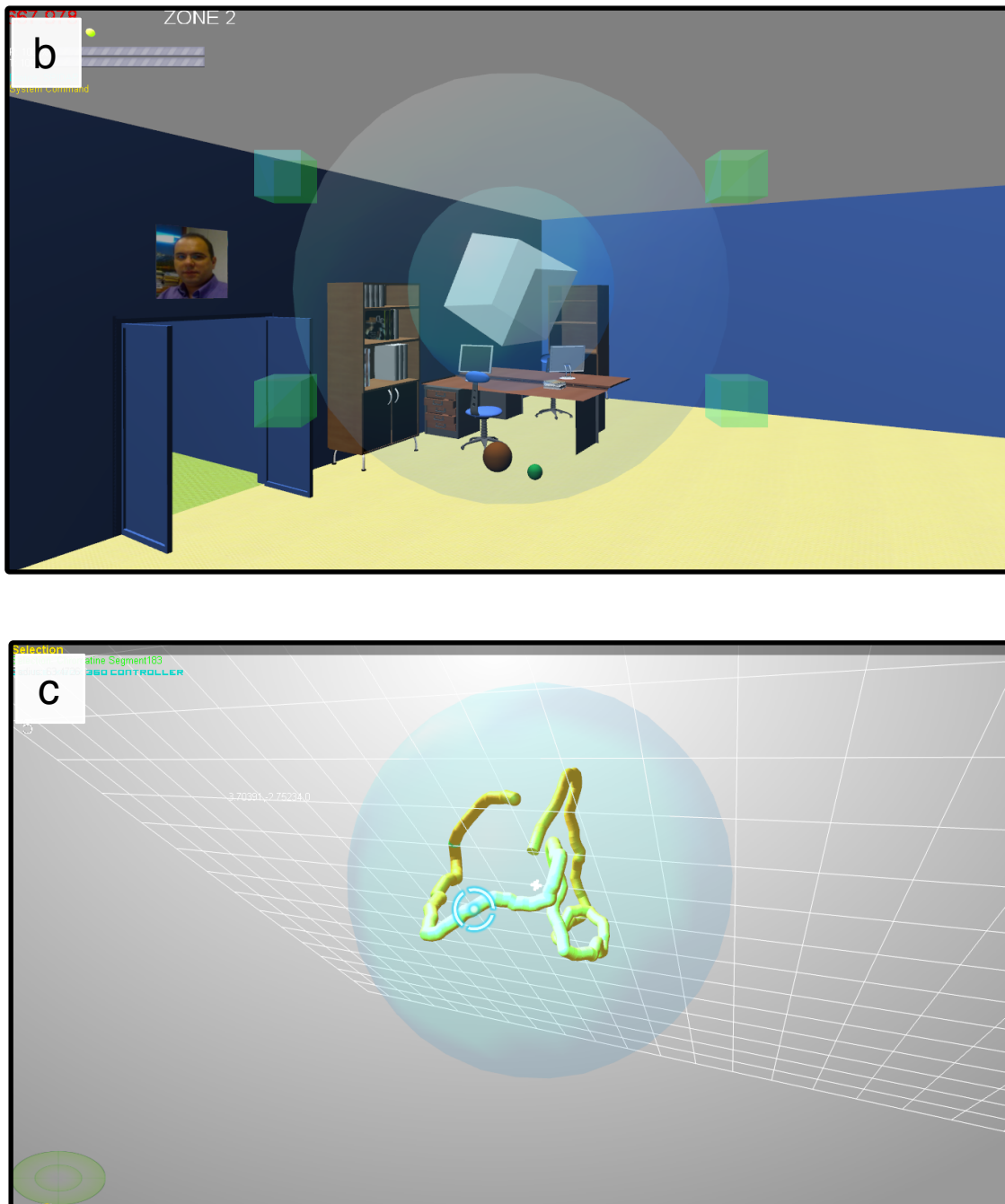


Figure 97: Différentes applications de réalité virtuelle utilisant la technique d'interaction Fly-Over.

4.7.2 Conclusion

Nous avons pu voir avec ce chapitre l'ensemble de la solution que nous avons proposée afin de permettre à une personne d'interagir en 3D de manière continue. Cette solution n'a pas été seulement la conception d'une bonne technique d'interaction, mais un ensemble complet de mesures, prises sur tous les aspects d'un système de réalité virtuelle, en partant de la couche matérielle et de la gestion du flot de données, jusqu'à la partie logicielle comportant une technique d'interaction 3D innovante et des assistances

à l'interaction 3D pour l'utilisateur. La figure 98 résume les différentes couches que nous proposons pour notre système d'interaction 3D adaptative, chacune correspondant à la réponse à un besoin spécifique.

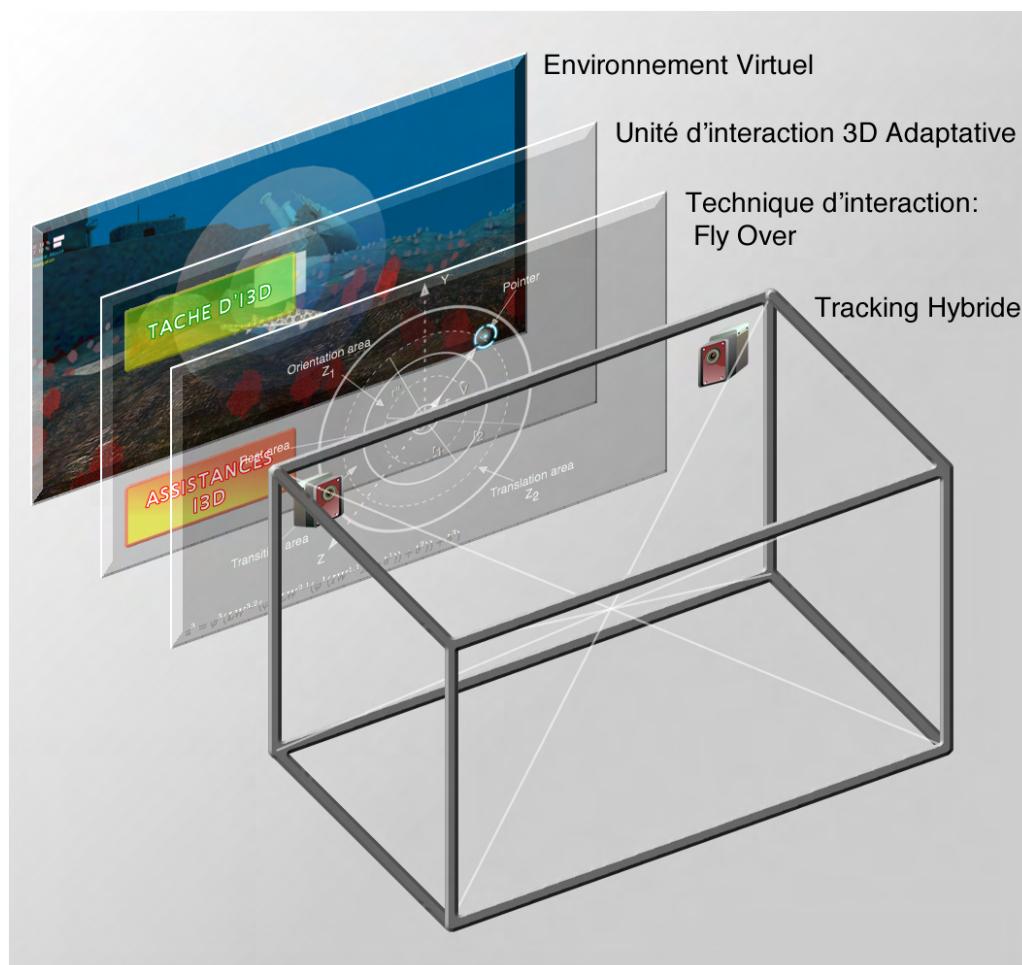


Figure 98: Les différentes couches de notre système d'interaction 3D adaptative permettant à une personne d'interagir en continu.

Ainsi, on distingue trois couches, pour :

1. Proposer une métaphore pour la technique d'interaction 3D permettant un apprentissage rapide et qui soit la même pour toutes les tâches de l'interaction 3D;
2. Passer de l'une à l'autre de manière continue et de conserver la même logique d'utilisation pour chacune des tâches;
3. Permettre la continuité des données à l'aide d'un système de tracking hybride garantissant un vecteur position sans discontinuité. C'est ce que nous allons présenter dans le chapitre suivant.

Pour le répondre au premier point, nous avons conçu et modélisé la technique d'interaction 3D Fly-Over, qui permet de réaliser toutes les tâches de l'interaction 3D en gardant la même logique d'utilisation. Trouver une solution pour le second point a été grandement facilité par la conception même de Fly-Over. Ainsi, nous avons constaté que par l'observation de certains paramètres, on peut aisément prédire vers quelle tâche de l'interaction 3D l'utilisateur souhaité basculer. Nous avons donc vu que pour créer une

interface qui soit agréable pour l'utilisateur, il est crucial de trouver l'équilibre entre anticiper et interpréter l'intention de l'utilisateur, et lui laisser assez de liberté pour réaliser des tâches qui sortent du cadre qu'on avait initialement prévu. C'est là toute la difficulté, car pour peu qu'on sorte des rails, ce qui était appréciable et agréable en devient un véritable parcours du combattant, rendant plus compliqué ce qui avait vocation à devenir plus simple. En matière d'intelligence informatique, le *quiproquo* peut faire des ravages. Solutionner le dernier point n'a cependant pas encore été abordé, nous avons seulement présenté son outil : l'unité de correction et de combinaison de données. C'est ce que nous allons voir dans le chapitre suivant.

ETUDE ET CALIBRATION DU SPIDAR

5.1 INTRODUCTION

Ce chapitre traite de la caractérisation du SPIDAR ainsi que de l'ensemble des mesures prises afin d'estimer une position plus précise à l'aide de ce périphérique, c'est-à-dire procéder à la calibration du SPIDAR. Il aborde aussi la conception d'un tracking hybride basé sur le système de tracking optique et le SPIDAR. Cet aspect de la thèse est primordial, car elle garantit la continuité de l'interaction 3D, au travers de la gestion de la continuité des données issues du système de tracking.

Le principal périphérique utilisé dans la plupart des systèmes de réalité virtuelle, ainsi que dans le nôtre, est le tracking optique ART. C'est un dispositif très précis, en revanche il souffre d'un défaut majeur : la perte du suivi. Cela est d'autant plus vrai lorsque l'on utilise un seul marqueur. Il est donc primordial dans ces situations assez fréquentes de pouvoir basculer sur un autre système de tracking : celui fourni par le SPIDAR. Le SPIDAR, quant à lui, souffre d'un problème de précision et de répétabilité, il est donc impossible de l'utiliser tel quel dans notre système. En effet, le moindre saut de position lors du basculement d'un système, est susceptible de faire passer le pointeur de Fly-Over d'une zone d'interaction vers une autre. Ce phénomène entraîne ainsi un comportement de la technique, non souhaité par l'utilisateur et créant par conséquent une rupture de la continuité de l'interaction 3D. Il est donc important de proposer une solution afin d'estimer la position donnée par le SPIDAR afin qu'elle soit la plus proche de la position procurée par le tracking optique, minimisant ainsi les effets sur l'interaction 3D.

5.2 CARACTÉRISATION DU SPIDAR À L'AIDE DU TRACKING OPTIQUE

5.2.1 *Contexte*

Dans la réalisation d'un système permettant la continuité de l'interaction 3D, il est nécessaire de pallier un problème courant dans les systèmes de tracking, le saut de données. Celui-ci apparaît lorsqu'il y a une perte de tracking, puis de nouveau une disponibilité du tracking. En effet, durant ce laps de temps durant lequel le système de tracking ne nous informe plus sur le suivi, l'objet réel, lui, continue sa course. Dès que le système arrive à effectuer de nouveau le suivi, les données qu'il vient d'acquérir peuvent être plus ou moins différentes des dernières données acquises, engendrant ainsi un saut de données. Un saut de données peut être géré de trois manières différentes :

1. *Prédiction*: Nous pouvons prédire l'état suivant des données grâce à l'utilisation de méthodes adaptées (ex : filtre de Kalman).
2. *Palliation*: La perte de tracking d'un périphérique, ne nous empêche pas d'en utiliser un autre en attendant sa disponibilité. Dans ce cas-là, il est important de s'assurer que les données des deux périphériques de tracking se situent bien dans un espace de représentation commun et que le passage de l'un à l'autre se réalise sans saut de données notable.
3. *Correction*: Nous pouvons corriger la position donnée par le SPIDAR à l'aide d'un apprentissage précédemment effectué (ex : réseau de neurones) ou bien au vu de la connaissance à priori de la position donnée par un capteur sûr.

5.2.2 Problèmes de conception rencontrés dans le SPIDAR

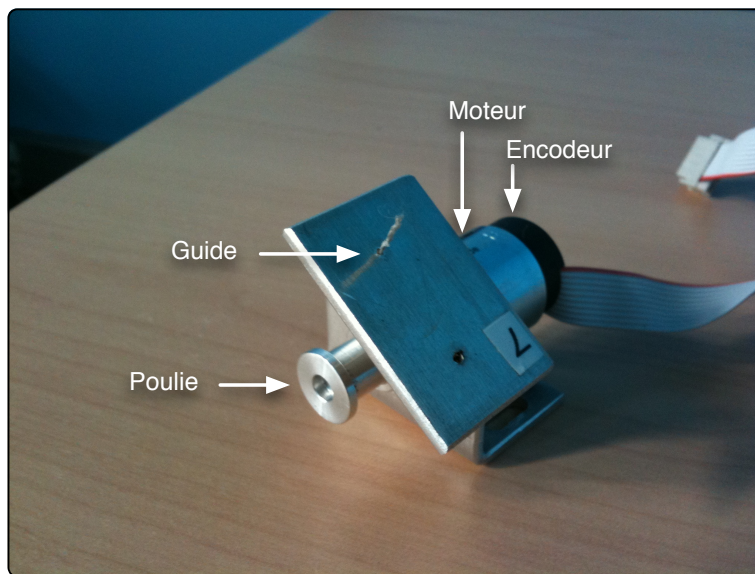


Figure 99: Vue détaillée d'un moteur du SPIDAR et du guide d'enroulement.

Le SPIDAR est un périphérique électromécanique, et par conséquent il peut souffrir de problèmes de conception pénalisant le calcul de la position de l'effecteur. Voici les problèmes de conception que nous avons identifiés :

- *Les encodeurs sont directement fixés sur l'axe moteur* : Le SPIDAR est équipé de moteurs avec encodeurs intégrés directement sur l'axe moteur. Cela est un problème car nous devons spécifier le diamètre du réa de la ficelle dans le fichier de configuration dont se sert le SPIDAR pour estimer la position de l'effecteur. Or, le réa est fixé sur l'axe moteur également, ce qui a pour effet de faire varier le diamètre du réa en fonction de la quantité de ficelle enroulée. Cette information est donc biaisée.
- *Le diamètre des réas* : Le diamètre des réas n'est pas adapté à la longueur de chaque ficelle. Dans la situation où une quantité importante de ficelle est enroulée par le réa, le diamètre de celui-ci varie de manière assez importante. Cette variation est d'autant plus importante que leur diamètre est petit.
- *Guide d'enroulement mal conçu* : Le SPIDAR ne possède pas non plus de guide d'enroulement qui empêche la ficelle de dérailler de la gorge du réa. En effet, nous

avons observé à plusieurs reprises ce phénomène, qui se réalise lorsque l'effecteur doit se déplacer de manière assez abrupte et que les moteurs doivent donc enrouler assez vite une grande quantité de ficelle. L'encodeur comptabilise un tour alors que la ficelle n'a pourtant pas été enroulée. Là encore, cette information est biaisée.

- *Taille des encodeurs* : les encodeurs chargés de compter le nombre de tour de réa ont une taille trop petite, donc ils débordent, c'est à dire que le compte repart à zéro, faussant ainsi le calcul de la quantité de ficelle enroulée.
- *Les dimensions du SPIDAR* : tous les problèmes précédemment cités sont accentués par les grandes dimensions de notre SPIDAR. Certains problèmes qui ont une influence faible lorsque le SPIDAR a de petites dimensions deviennent non négligeables lorsque celui-ci a des dimensions plus importantes.

Le SPIDAR est un périphérique qui utilise une librairie de programmation pour communiquer avec l'ordinateur. Mais cette librairie nous permet seulement de récupérer certains paramètres du SPIDAR mais en aucun cas nous ne pouvons agir sur le modèle permettant de calculer la position de l'effecteur. L'interface est une boîte noire. C'est pour cette raison que nous avons décidé de caractériser le SPIDAR. Pour cela, nous avons besoin de connaître la position de l'effecteur donnée par le SPIDAR ainsi que sa position réelle : celle de référence. Pour cela nous avons pensé utiliser le tracking optique ART qui est une référence en matière de précision et de fiabilité (erreur de l'ordre d'1 mm en position et du dixième de mm en orientation). Nous faisons donc faire une étude comparative des deux systèmes.

5.2.3 Hypothèses de travail

Afin de permettre une bonne caractérisation de ce périphérique, il est important de fixer quelques hypothèses de travail. Ainsi, nous restreignons cette étude à un espace de travail, plus petit que celui permis en théorie. D'un volume théorique de 27 m^3 nous passons à un volume de 3.375 m^3 (voir figure 100).

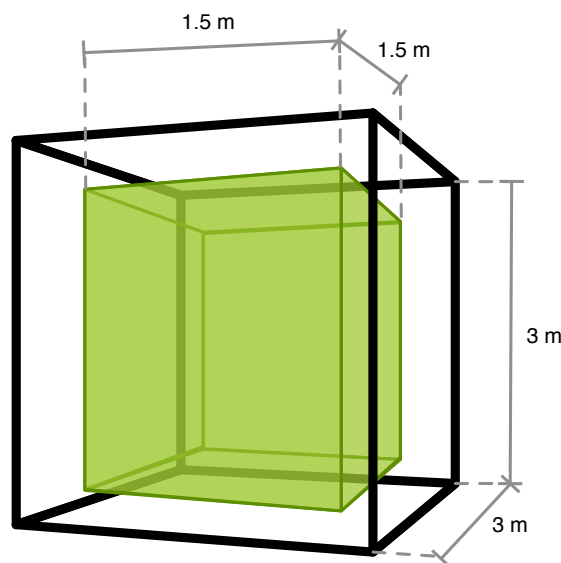


Figure 100: Représentation de l'espace de travail (en vert) du SPIDAR au sein de sa structure (en noir).

La raison de cette limitation est simple, elle nous permet, d'une part, de concentrer nos efforts sur la zone la plus utilisée et d'autre part de nous affranchir des situations dans lesquelles le SPIDAR fonctionne au bout de ses limites et qui, du coup, nous conduit à une mauvaise interprétation des résultats. Cette limitation n'est cependant pas contraignante, dans le sens où cette espace de travail couvre largement les besoins de l'utilisation du SPIDAR.

5.2.4 Protocole expérimental mis en place

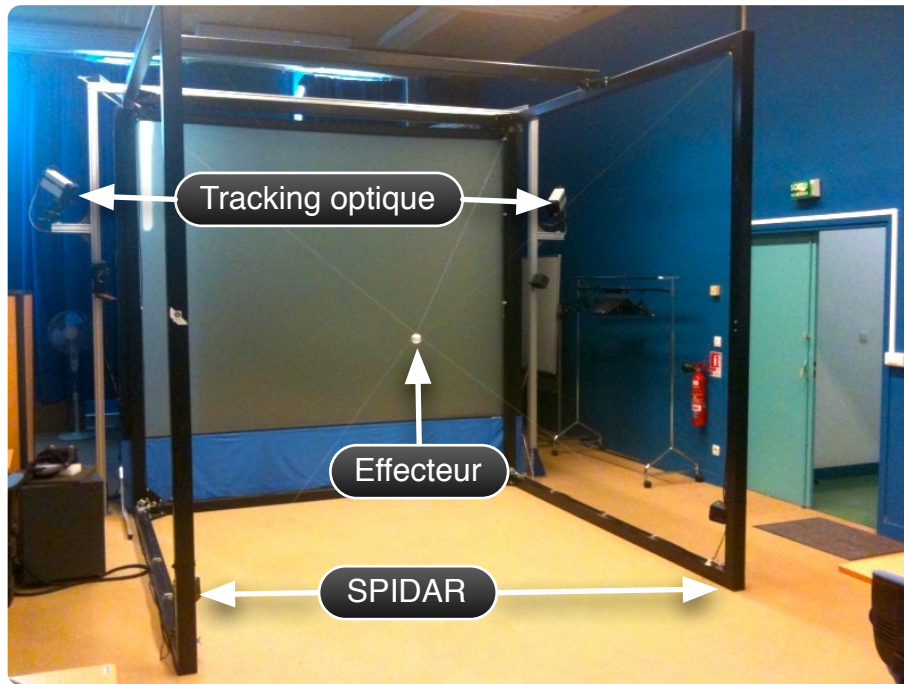


Figure 101: Disposition du matériel.

Ce protocole a été difficile à concevoir, dans la mesure où, comme nous avons pu le voir dans la section 3, il est très difficile de caractériser un système de tracking 3D. Cette difficulté provient essentiellement de contraintes techniques puisqu'il est impossible de contraindre les mesures à saisir à des positions précises. Avec ce protocole on utilise simultanément la position fournie par le SPIDAR et le tracking optique ART. Cela est possible au sein de notre plate-forme de réalité virtuelle : EVR@ [Laboratoire IBISC, 2010], par la disposition des deux caméras infrarouges du système de tracking optique de part et d'autre de la structure du SPIDAR (voir figure 101). Ce protocole consiste à représenter une scène virtuelle constituée de plusieurs petits cubes. Chaque cube correspond à un sous-espace de l'espace de travail du SPIDAR. L'ensemble de ces cubes délimite l'espace de travail du SPIDAR. On peut voir une représentation de ce protocole dans la figure 102.

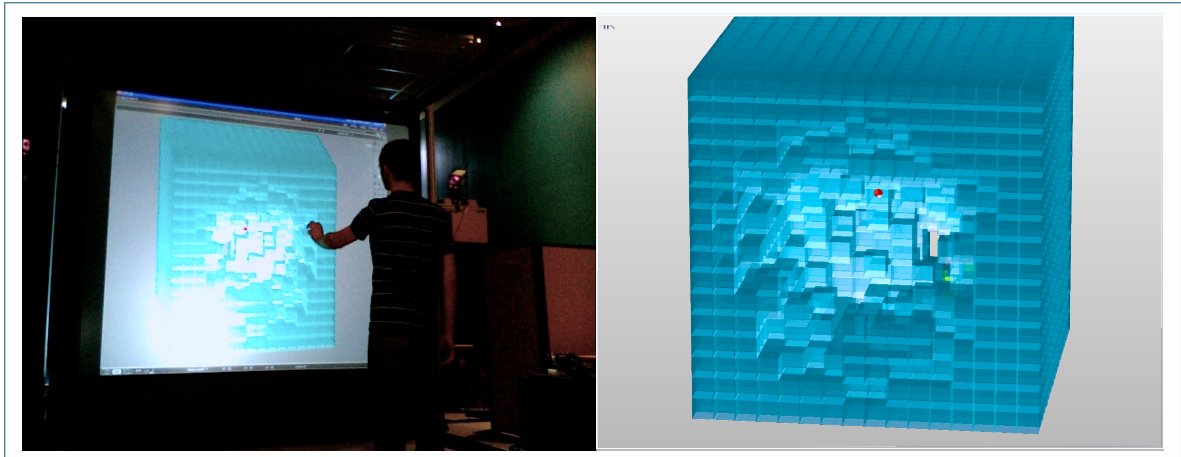


Figure 102: Cadre expérimental du protocole. A gauche, un utilisateur est en train de réaliser la prise des positions. A droite, une vue détaillée de ce que perçoit l'utilisateur (en rouge l'effecteur virtuel que contrôle l'utilisateur).

Ce protocole permet de guider l'utilisateur à placer l'effecteur dans l'espace de manière à récupérer un ensemble de positions qui seront réparties dans l'espace. Pour cela, l'utilisateur se sert de la représentation virtuelle de l'espace de travail. Ce protocole se déroule de la manière suivante :

1. L'utilisateur déplace l'effecteur dans l'espace afin de placer un effecteur virtuel dans chacun des cubes.
2. A chaque fois que l'effecteur virtuel est en collision avec un cube, on enregistre la position donnée par le SPIDAR, la position donnée par le système ART ainsi que les valeurs provenant des différents encodeurs.
3. Une fois ces données récupérées, le cube disparaît, garantissant ainsi l'unicité des valeurs de la zone couverte.

Le grand avantage de ce protocole est qu'il permet de couvrir un espace de travail contrôlé et fixé par l'expert ainsi que d'avoir une répartition homogène dans l'espace des mesures. Nous avons relevé des mesures en respectant ce protocole avec une zone de travail limitée à 1m^3 qui est décomposé en 4096 sous-espaces. Ce protocole est d'autant plus original, qu'il utilise les possibilités de la réalité virtuelle afin de répondre à un problème réel.

5.2.5 Incertitudes sur la position du SPIDAR

Comme nous pouvons le voir dans la figure 103, la répartition de l'erreur absolue dans l'espace de travail du SPIDAR est de type "pelure d'oignon", c'est-à-dire différentes couches, plutôt sphériques, l'erreur absolue grandissant au gré des couches. Les zones où l'erreur sur la position est la plus importante, sont situées sur le dessus et les différents coins de l'espace de travail.

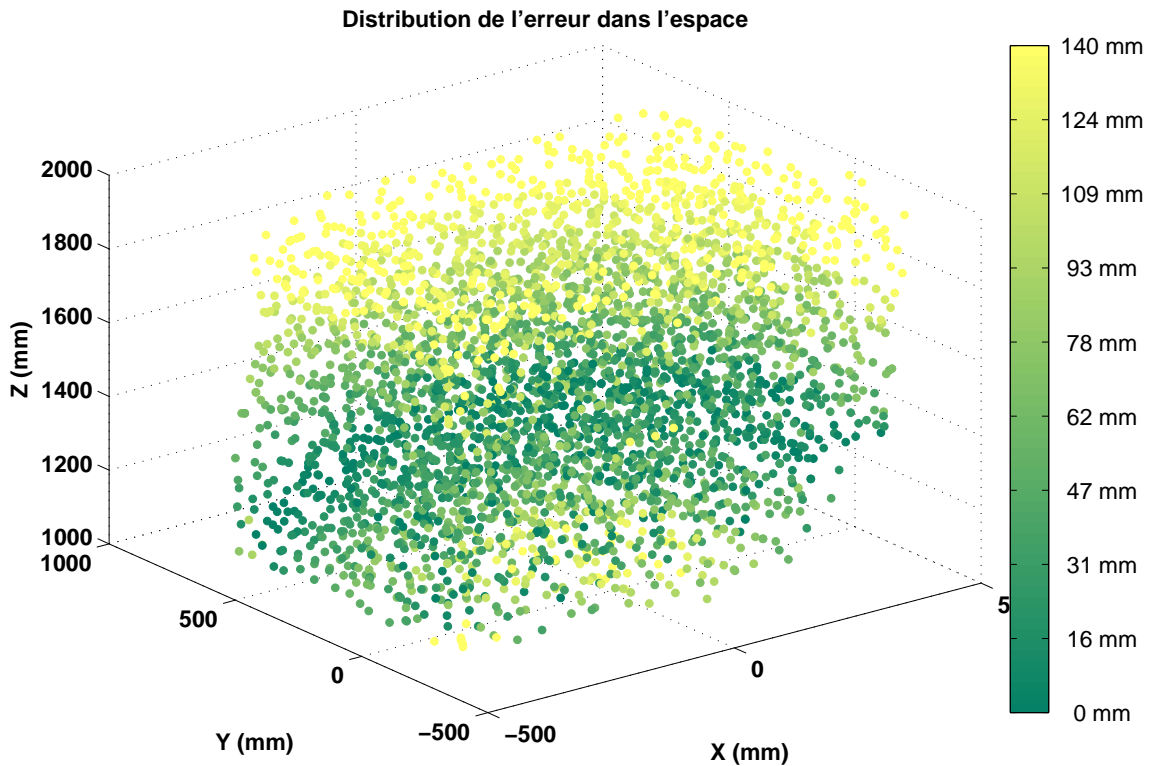


Figure 103: Représentation de l'erreur absolue dans le volume de travail du SPIDAR.

CONCLUSION DE LA CARACTÉRISATION

Cette caractérisation nous a permis d'identifier différents problèmes :

- L'étude mécanique démontre qu'il y a trop de problèmes de conception;
- Il est difficile de quantifier la portée de ces différents problèmes mécaniques;
- Le manque de connaissance sur le modèle de calcul utilisé par l'interface du SPIDAR afin d'estimer la position de l'effecteur;
- Il faut rajouter une API qui mérite d'être mieux documentée.

Finalement, le SPIDAR souffre d'un ensemble de problèmes dont nous ne faisons que soupçonner la cause, et pour lesquels nous avons du mal à évaluer l'influence de chacun sur l'estimation de la position finale de l'effecteur. Pour toutes ces raisons, l'utilisation d'une approche permettant d'estimer des données sans avoir de connaissance a priori sur son modèle mathématique, semble être la solution afin d'améliorer la précision du SPIDAR sur la position qu'il fournit.

5.3 INITIALISATION MULTIMODALE DU SPIDAR

5.3.1 Contexte

Le SPIDAR est un périphérique nécessitant une initialisation à chaque mise sous tension. Cette initialisation consiste à définir le nouveau référentiel utilisé par le SPIDAR. Pour l'effectuer, il faut placer l'effecteur du SPIDAR au centre de sa structure

avec la plus grande précision possible. Une mauvaise initialisation entraîne une baisse des performances du SPIDAR. De plus, si celle-ci n'est pas quasi-identique à chaque mise sous tension, cela nous met dans l'incapacité d'appliquer un traitement sur les données du SPIDAR, puisque tout traitement dépend du référentiel utilisé. Il est donc important de procéder à cette initialisation avec une grande attention. Le problème provient justement du placement de l'effecteur. Il est en effet très difficile de placer l'effecteur au bon endroit, avec une bonne précision. La principale cause étant que l'utilisateur ne dispose d'aucun repère pour estimer le centre de l'espace de travail du SPIDAR.

5.3.2 Solution proposée pour minimiser l'influence des problèmes de conception

Nous proposons une initialisation multimodale du SPIDAR, c'est-à-dire que l'utilisateur est guidé pour effectuer l'initialisation par le biais de plusieurs modalités, qui sont visuelles, sonores et haptiques. Pour effectuer l'initialisation le plus justement possible, nous avons eu besoin de déterminer ce point précisément dans l'espace. Pour cela, nous avons eu l'idée d'utiliser le système de tracking optique de la plate-forme, qui nous offre une précision en position de l'ordre du millimètre.

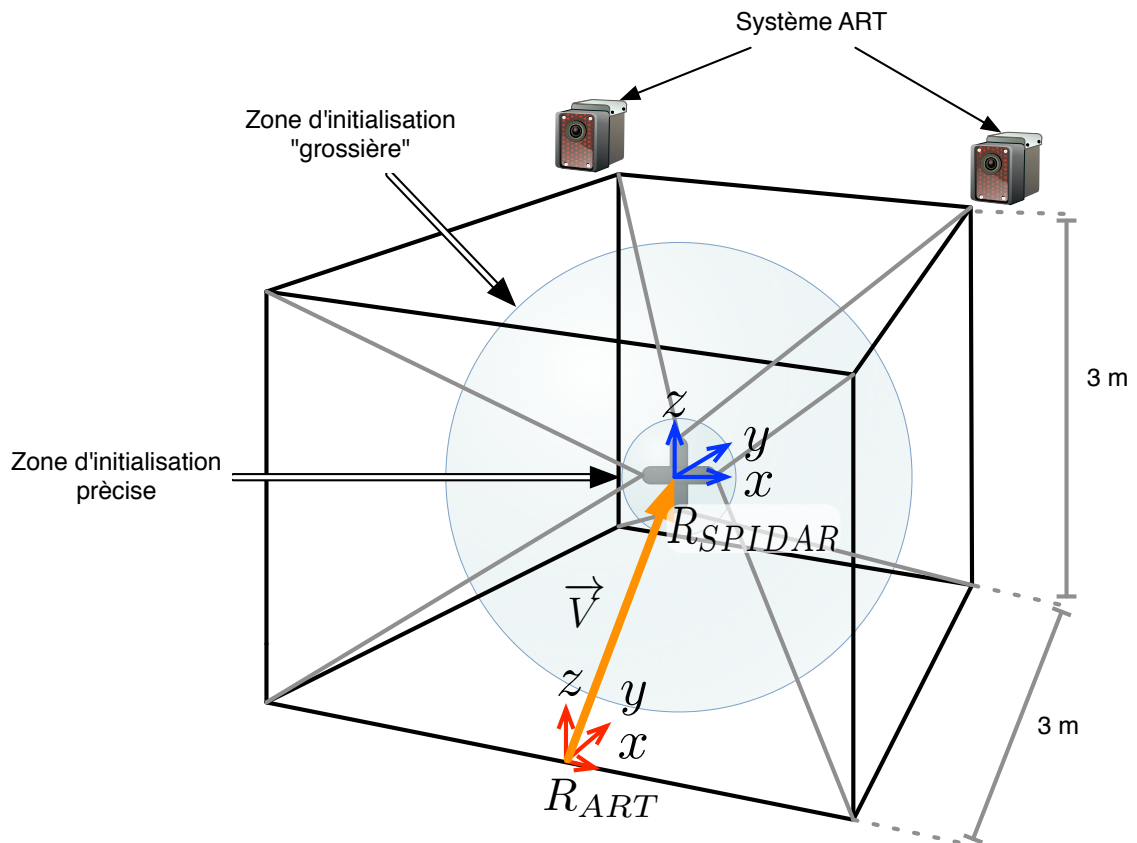


Figure 104: Zones d'initialisation et cadre expérimental.

Nous connaissons la disposition géométrique des caméras infrarouges, de l'origine du repère du tracking optique ainsi que la disposition géométrique de la structure du SPIDAR. Nous avons donc pu déterminer la position dans l'espace du point d'initialisation

théorique du SPIDAR. Une fois cette étape achevée, il a fallu trouver un moyen de connaître la position de l'effecteur du SPIDAR dans le repère ART. Pour cela, nous avons fixé un marqueur optique sur l'effecteur, ainsi le système ART peut connaître la position de l'effecteur dans son repère. Donc, en calculant le vecteur défini par le point d'initialisation théorique et l'effecteur, nous pouvons déterminer la distance à laquelle l'effecteur se situe par rapport au point d'initialisation ainsi que la direction à prendre pour y aller.

A - PARAMÈTRES MULTIMODAUX

Afin de guider l'utilisateur pour l'initialisation, il a fallu convertir ces informations en paramètres multimodaux. On a donc défini une fonction h , représentative de la distance d à laquelle on se trouve du point d'initialisation. Ainsi, plus $h(d)$ tend vers 1 et plus on est proche du point d'initialisation. A l'opposé si $h(d)$ tend vers 0 on s'en éloigne. On a défini $d_{\min} = 1000\text{mm}$ qui représente la distance minimale à laquelle la fonction $h(d)$ commence à varier, sinon $h(x)$ est tout simplement égal à zéro.

$$h(d) = \begin{cases} h(d) = \frac{d_{\min} - d}{d_{\min}} & \text{si } d \leq d_{\min} \\ h(d) = 0 & \text{si } d > d_{\min} \end{cases} \quad (5.1)$$

$$d = \|\vec{V}\| \quad \text{avec} \quad \vec{V} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \quad (5.2)$$

Les paramètres multimodaux sont les suivants :

- Modalité sonore : la distance permet de déterminer la fréquence à laquelle des signaux sonores sont diffusés (à la manière d'un radar de recul sur une automobile).

$$f(d) = h(d) \cdot 1000 \quad \text{si } d \leq d_{\min} \quad (5.3)$$

- Modalité haptique : la direction du vecteur est convertie en vecteur force appliqué sur l'effecteur afin d'indiquer au système d'initialisation dans quelle direction déplacer l'effecteur afin de le faire converger vers le point d'initialisation.

$$\vec{F} = \vec{\Delta} \cdot s(d) \cdot F_{\max} \quad (5.4)$$

Avec :

$s(x)$, la fonction sigmoïde permettant d'appliquer une force progressive sur l'effecteur, définie telle que :

$$s(x) = e^{-10((1-h(d))-0.5)^2} \quad \text{si } x \in [-0.5; 0.5] \quad (5.5)$$

$\vec{\Delta}$, le vecteur unitaire dirigé vers le point d'initialisation c'est-à-dire la normalisation du vecteur \vec{V} :

$$\vec{\Delta} = \frac{1}{\|\vec{V}\|} \cdot \vec{V} \quad (5.6)$$

- Modalité visuelle : la position relative de l'effecteur par rapport au point d'initialisation permet d'indiquer visuellement si ces deux points sont proches ou pas.

B - ALGORITHME D'INITIALISATION DU SPIDAR

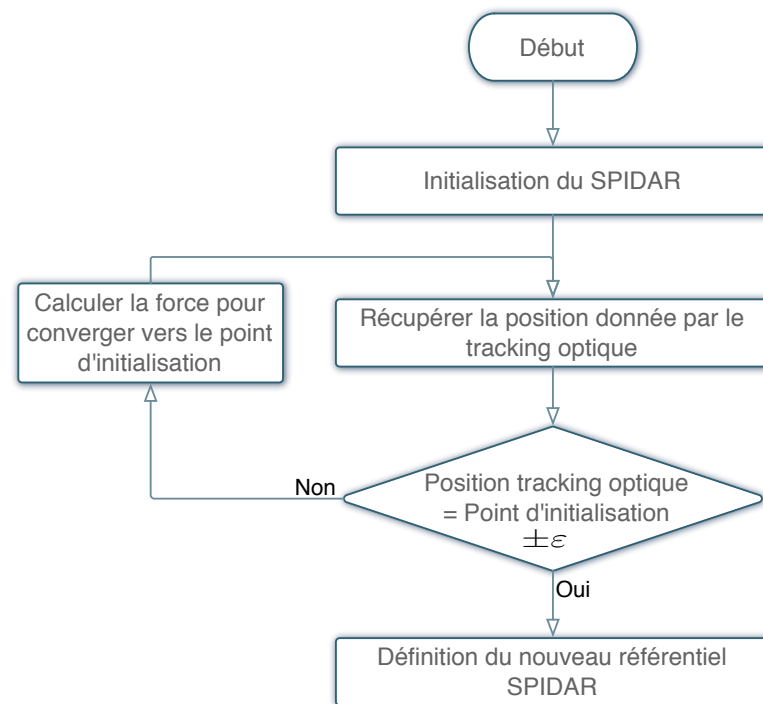


Figure 105: Algorithme du processus d'initialisation du SPIDAR.

Une première initialisation dite grossière permet de placer l'effecteur du SPIDAR relativement près du point d'initialisation. Avec cette première initialisation, nous arrivons à une précision de l'ordre de 1.5 cm. En revanche avec cette méthode, nous ne pouvons atteindre une meilleure précision. En effet, le déplacement de l'effecteur, par l'application d'une force, ne peut se faire de manière précise car le SPIDAR ne dispose pas d'une sensibilité suffisante, c'est pour cette raison que nous avons dû utiliser une deuxième phase, dite d'initialisation précise. Une fois l'initialisation grossière effectuée, il nous faut nous rapprocher encore plus près du point d'initialisation. Afin d'y parvenir, nous avons utilisé une autre technique, qui consiste à guider haptiquement l'utilisateur afin de le faire converger vers le point d'initialisation, dans une zone restreinte autour du point d'initialisation. Ainsi nous avons statistiquement une chance d'atteindre encore plus précisément le point d'initialisation. Avec une telle méthode nous arrivons à une précision de l'ordre de 1.2 mm. L'ensemble du processus d'initialisation est résumé par l'algorithme représenté figure 105.

5.4 CALIBRATION DU SPIDAR PAR UNE RÉGRESSION LINÉAIRE

5.4.1 Présentation

En mathématiques et plus précisément en analyse numérique, l'interpolation polynomiale est une technique d'interpolation d'un ensemble de données ou d'une fonction par un polynôme. En d'autres termes, étant donné un ensemble de points (obtenu, par exemple, à la suite d'une expérience), on cherche un polynôme qui passe par tous ces

points, et qui éventuellement vérifie d'autres conditions, de degré si possible le plus bas. Le résultat n'est toutefois pas toujours à la hauteur des espérances : l'interpolation de Lagrange, par exemple, peut fort bien diverger même pour des fonctions très régulières (phénomène de Runge).

Par cette méthode, nous obtenons un ensemble de trois fonctions linéaires, permettant de transformer les composantes (x_o , y_o et z_o) de chacun des vecteurs positions du SPIDAR (P_o) sur les 3 axes x , y et z définies par :

$$f\left(\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}\right) = \begin{bmatrix} 0.7201 \\ 0.7286 \\ 0.7571 \end{bmatrix} \cdot \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}^T + \begin{bmatrix} 44.0105 \\ 7.9741 \\ 337.6187 \end{bmatrix} \quad (5.7)$$

5.4.2 Résultats obtenus par une régression linéaire (interpolation polynomiale d'ordre 1)

La figure 106 représente la variation des erreurs absolues sur la position du SPIDAR avant et après calibration par interpolation polynomiale d'ordre 1, en fonction de la distance à laquelle l'effecteur est situé du point d'initialisation. Dans le cas idéal, une ligne droite horizontale d'équation $f(x) = 0$ est représentée, c'est-à-dire qu'il n'existe aucune erreur sur la position et que celles-ci ne varient pas en fonction de la distance du point d'initialisation. Or, comme nous pouvons le voir, les erreurs absolues varient de manière non linéaire avant calibration, ce qui confirme l'interprétation du graphique 103. Après calibration du SPIDAR par l'interpolation polynomiale d'ordre, on observe une nette amélioration de la précision (le nuage de points est plus écrasé). Cependant la distribution des erreurs se faisant de manière non linéaire et cette méthode de calibration étant une approche linéaire, l'amélioration n'est que partielle dans des zones où le comportement du SPIDAR est quasi-linéaire.

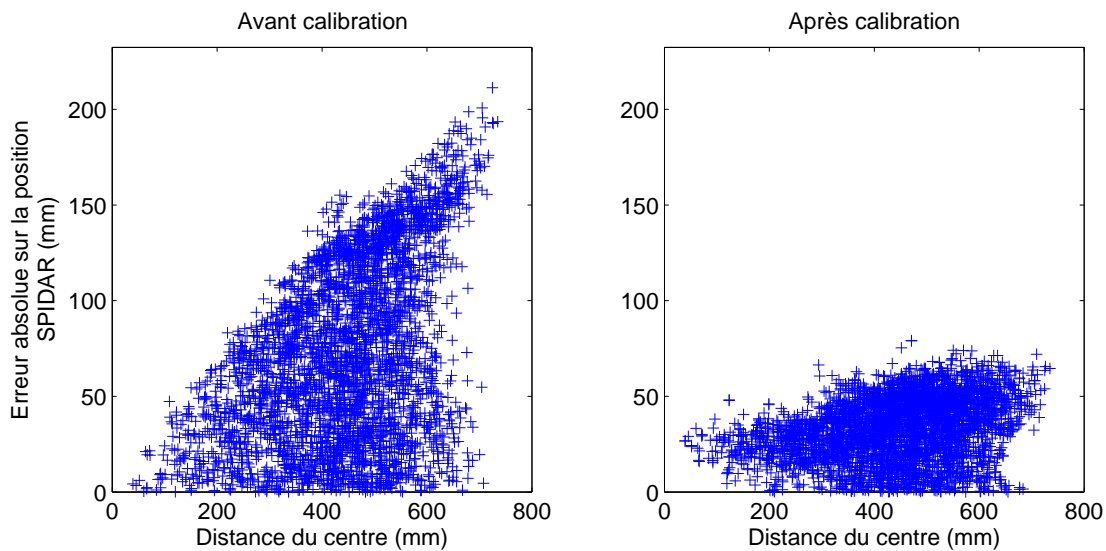


Figure 106: Erreurs absolues sur la position du SPIDAR par rapport à la distance à laquelle la mesure a été prise du point d'initialisation, avant et après calibration par interpolation polynomiale d'ordre 1.

5.4.2.1 Répartition de l'erreur absolue sur la position dans l'espace de travail du SPIDAR.

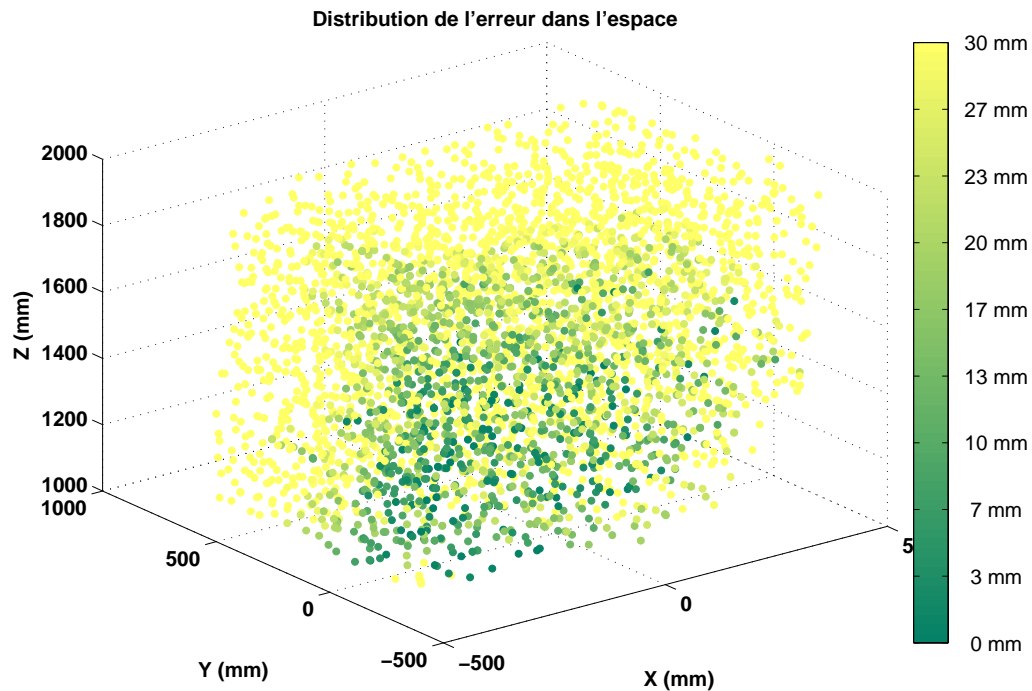


Figure 107: Répartition de l'erreur absolue sur la position du SPIDAR dans l'espace de travail.

Comme le montre la figure 107, une grande partie de l'espace de travail comporte encore des erreurs supérieures ou égales à 30 mm sur la position après calibration (points de couleur claire). Seule une zone située entre le centre et le bas de l'espace de travail comporte des erreurs inférieures ou égales à 15 mm (points de couleur foncée).

5.4.2.2 Histogrammes des erreurs

La figure 108 et le tableau 3 représentent respectivement la distribution des erreurs absolues en fonction des plages d'erreurs avant (brut) et après calibration par une régression linéaire (IP1) et les valeurs caractéristiques de ces erreurs. Avant calibration par régression linéaire, nous sommes face à un histogramme plutôt étalé, centré sur 73 mm et avec un écart-type de 47 mm, témoignant d'une mauvaise précision et d'une grande dispersion statistique (l'idéal étant un histogramme en pointe centré sur 0 mm). Après calibration (IP1), l'histogramme est moins étalé, on distingue une gaussienne centrée sur une valeur proche de 30 mm et une forte diminution de l'écart-type (16 mm). On a donc une nette amélioration de la précision ainsi qu'une réduction de la dispersion statistique.

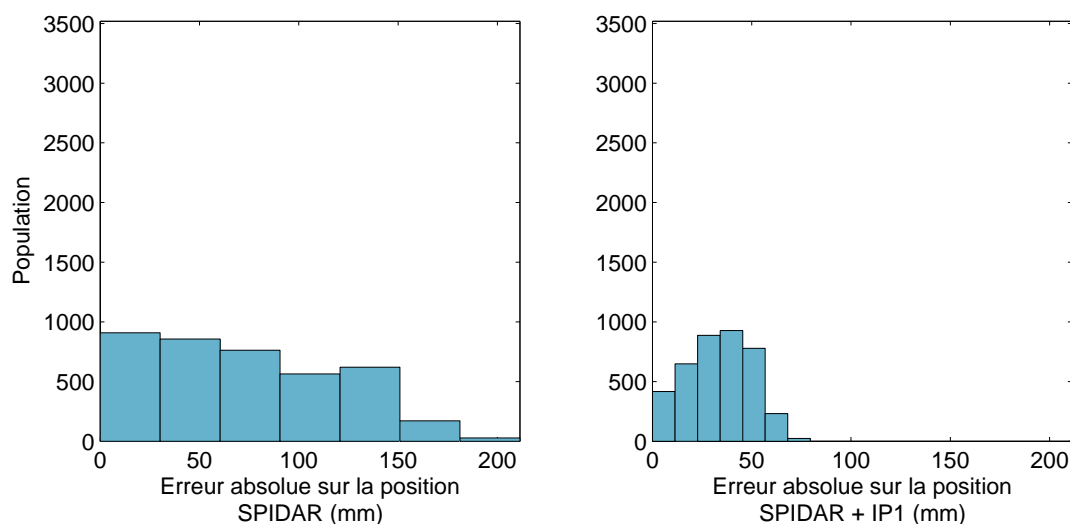


Figure 108: Erreur absolue sur la position du SPIDAR.

ERREUR ABSOLUE	BRUT	IP1
MOYENNE (MM)	72.86	33.53
ECART-TYPE (MM)	47.06	16.01
MAX (MM)	211.29	79.51

Table 3: Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par régression linéaire.

5.4.2.3 Conclusion sur l'utilisation d'une méthode de régression linéaire

Les résultats obtenus par le biais de cette méthode ne sont pas satisfaisants, car comme le montre le tableau 3, il existe toujours une erreur absolue qui avoisine 30 mm. De plus, à la vue de la répartition de l'erreur en 3D dans l'espace de travail, on distingue très nettement que seule une partie de l'espace de travail est corrigée. Cela paraît logique, dans la mesure où la répartition des erreurs dans l'espace avant correction (voir figure 104), montre bien que le système n'est pas linéaire. Il est donc tout à fait normal qu'en utilisant une méthode de régression linéaire, nous ne parvenions pas à obtenir une répartition homogène de l'erreur absolue dans l'espace. Cependant on notera que l'écart-type a diminué de plus de moitié en passant d'environ 47 mm à presque 16 mm, ce qui signifie que les distorsions ont diminué.

5.5 CALIBRATION DU SPIDAR À L'AIDE D'UN RÉSEAU DE NEURONES

5.5.1 Présentation

À la lumière des résultats précédents, nous avons jugé que l'utilisation d'un réseau de neurones peut être pertinente en tant que solution de correction de la position du

SPIDAR. Nous invitons le lecteur à consulter la partie 3.4.2.1, pour une présentation détaillée des réseaux de neurones.

5.5.2 Configuration du réseau de neurones

Nous avons paramétré le réseau de neurones en deux couches. La première couche avec une fonction d'activation de type sigmoïde tangente. La seconde avec une fonction d'activation de type rampe. Le nombre de neurones sur la couche cachée est trouvé de manière empirique. Cette démarche est expliquée dans la section 5.5.4. Le réseau de neurones est de type rétro-propagation et utilise l'algorithme d'apprentissage Levenberg-Marquardt. L'algorithme d'apprentissage incrémental du réseau est la descente de gradient à taux d'apprentissage fixe avec moment. L'erreur quadratique moyenne est employée comme fonction de coût.

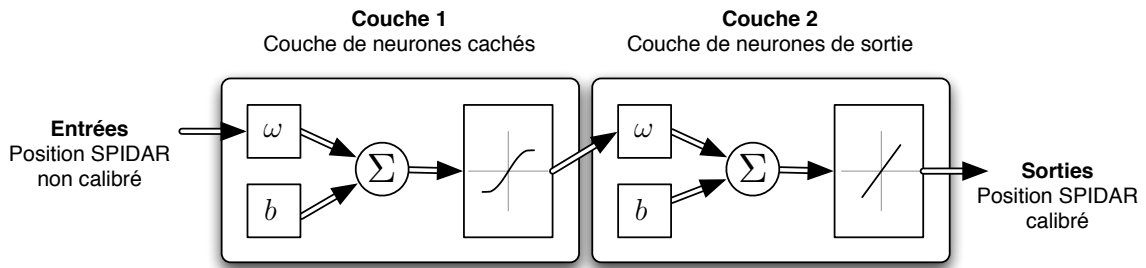


Figure 109: Réseau de neurones utilisé pour la correction du SPIDAR.

5.5.3 Apprentissage du réseau de neurones

Pour la phase d'apprentissage, nous avons utilisé la position du SPIDAR, relevées lors des campagnes de mesures, comme données d'entrée; ainsi que les positions recueillies avec le système optique ART en données de sortie. Cependant toutes les données n'ont pas été prises en compte, seules les données où le tracking optique est présent et fonctionnel ont été sélectionnées. Ceci afin d'avoir le meilleur apprentissage possible et d'écartier ainsi les données qui auraient faussé l'apprentissage du réseau de neurones. Mais avant cela, il nous faut normaliser ces données, ainsi :

$$\begin{cases} N_{spi} = ((P_{spi} - \Gamma_{spi}) \cdot \Sigma_{spi}) \\ N_{art} = ((P_{art} - \Gamma_{art}) \cdot \Sigma_{art}) \end{cases} \quad (5.8)$$

Où :

$$\begin{aligned} \Gamma_{spi} &= \gamma_{spi} \cdot J \quad \text{et} \quad \Sigma_{spi} = \sigma_{spi} \cdot J \\ \Gamma_{art} &= \gamma_{art} \cdot J \quad \Sigma_{art} = \sigma_{art} \cdot J \end{aligned} \quad (5.9)$$

Avec :

γ_{spi} : Matrice des moyennes des composantes de la matrice des positions recueillies avec le SPIDAR.

γ_{art} : Matrice des moyennes des composantes de la matrice des positions recueillies avec le système ART.

J : Matrice unitaire.

P_{spi} : Positions brutes du SPIDAR.

P_{art} : Positions brutes du tracking optique ART.

N_{spi} : Positions normalisées du SPIDAR.

N_{art} : Positions normalisées du tracking optique ART.

Les données recueillies avec chacun des protocoles sont ensuite découpées aléatoirement en 3 bases de données.

- La première (60% des données) sert pour l'apprentissage a proprement parler.
- La seconde (20% des données) est utile pour la validation de l'apprentissage et ainsi, vérifier qu'il n'y ait pas de phénomène de sur-apprentissage.
- Enfin la dernière (20% des données) est employée pour la généralisation, c'est à dire, la présentation de données totalement inconnues (n'ayant pas servies à l'apprentissage) du réseau de neurones.

5.5.4 Nombre optimal de neurones dans la couche cachée

Le nombre de neurones nécessaire dans la couche cachée afin d'approcher la solution optimale a été déterminée de manière empirique, en testant l'apprentissage avec différents réseaux de neurones possédant un nombre de neurones différents et en identifiant le plus performant, c'est à dire celui qui nous donne l'erreur absolue moyenne en position la plus faible. La figure 110 représente l'erreur absolue moyenne sur la position corrigée du SPIDAR en fonction du nombre de neurones présents dans la couche cachée du réseau de neurones. La meilleure configuration du réseau de neurones parmi celles testées, allant de 3 à 21 neurones pour la couche cachée, est celle possédant 5 neurones. On obtient une erreur absolue moyenne sur la position réelle de 12.62 mm.

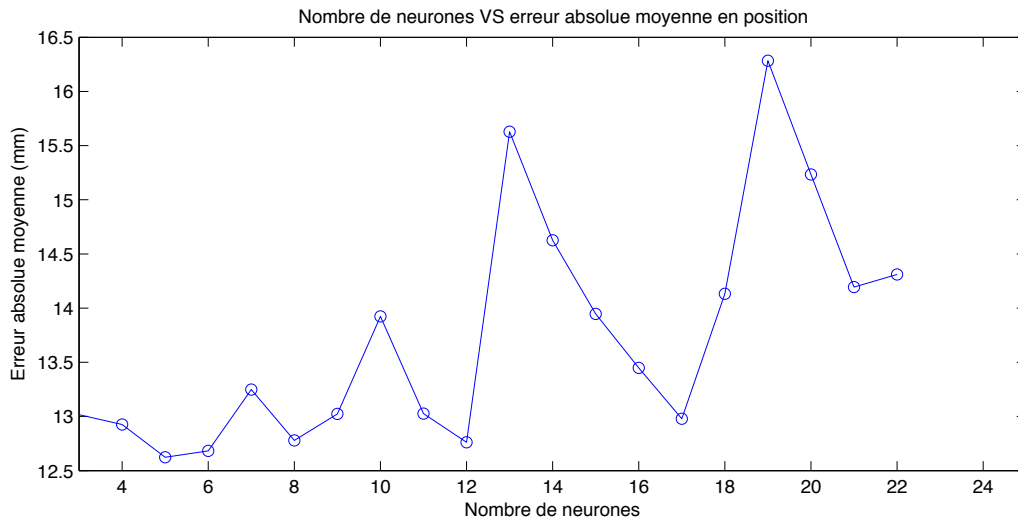


Figure 110: Erreur absolue moyenne sur la position par rapport au nombre de neurones sur la couche cachée.

La figure 111 représente l'évolution de la fonction de coût du SPIDAR tout au long de l'apprentissage. Cette courbe représente, l'erreur quadratique moyenne chutant lors

de la descente du gradient. On distingue très nettement une stabilisation de cette erreur après seulement quelques essais (epoch sur le graphique), cependant un des principaux inconvénients du réseau de neurones est l'incertitude d'atteindre le minimum global. En effet, il est possible que la fonction coût connaisse encore une ou plusieurs diminutions en continuant les essais.

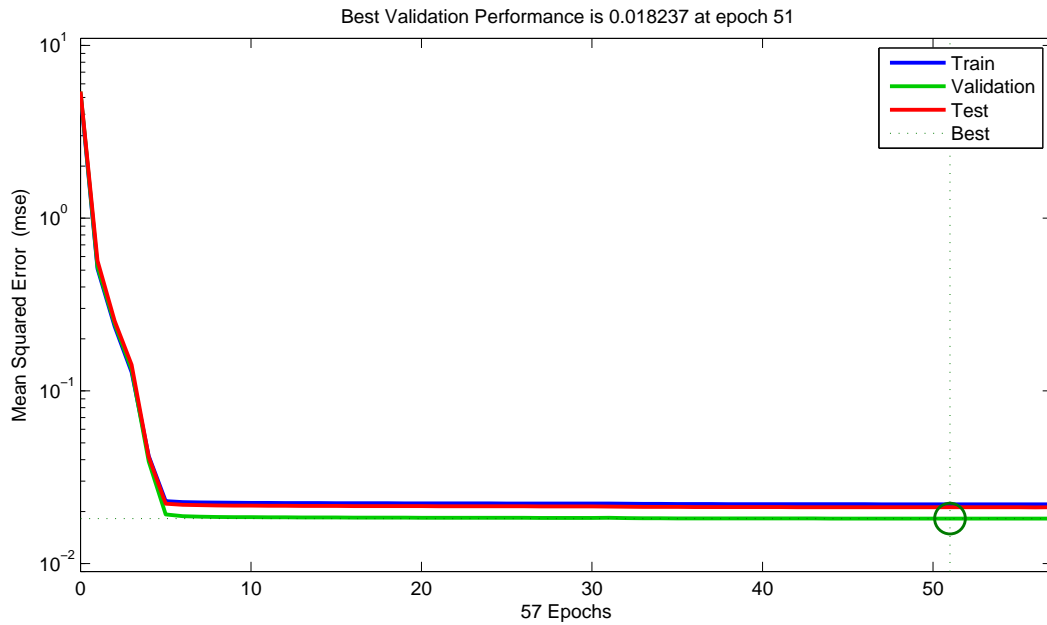


Figure 111: Performance du réseau de neurones utilisé pour la calibration.

Nous pouvons donc représenter graphiquement la configuration du réseau de neurones à rétro-propagation (figure 112) avec 5 neurones sur la couche cachée et 3 sur la couche de sortie.

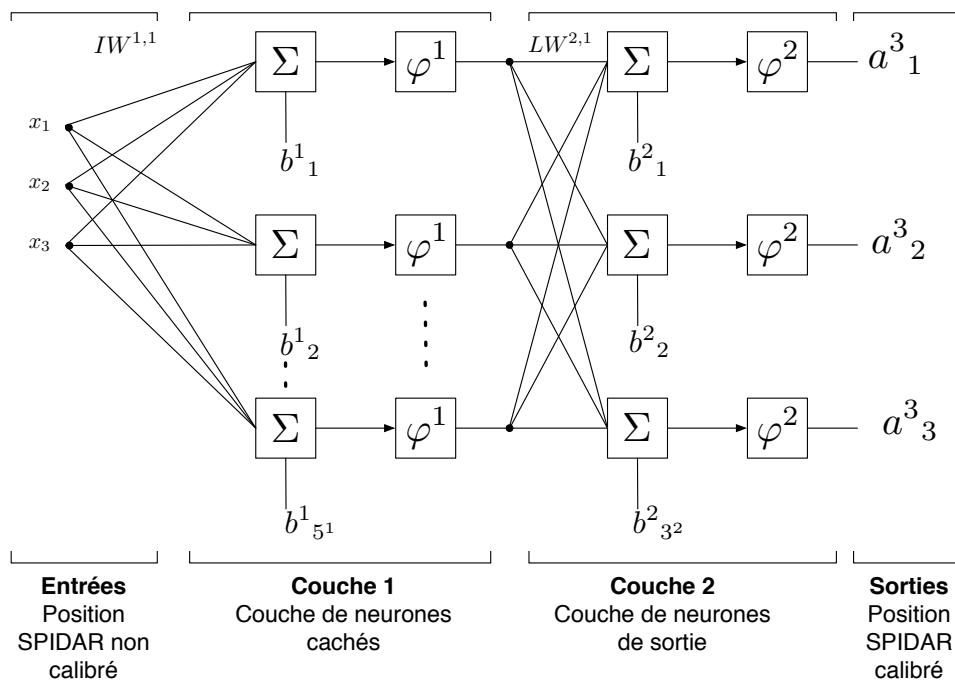


Figure 112: réseau de neurones utilisé pour la calibration du SPIDAR

5.5.5 Performances du réseau de neurones en apprentissage

Afin d'évaluer les performances du réseau de neurones, nous avons dû réfléchir au meilleur moyen de refléter celles-ci aux travers de résultats numériques et graphiques basés sur l'erreur absolue sur la position du SPIDAR par rapport à la position donnée par le tracking optique.

Nous avons donc choisi de nous appuyer sur quatre études, c'est-à-dire :

- L'histogramme des erreurs absolues sur la position;
- L'erreur moyenne absolue sur la position,
- L'écart-type sur les erreurs absolues,
- L'erreur absolue maximale;
- Un graphique en 3D représentant la distribution de l'erreur absolue dans l'espace.

La figure 113 représente la corrélation entre les données cibles, c'est-à-dire les positions données par le tracking optique, et les données en sortie du réseau de neurones, c'est-à-dire les positions données par le SPIDAR après calibration, pour les différentes bases de données : (apprentissage, validation, test et l'ensemble des données). Plus la fonction représentée s'apparente à une droite, sans points orphelins, plus le réseau de neurones est efficace. Pour chacune des bases de données, est donnée l'équation de la droite passant au plus près de l'ensemble des points du graphique concerné, l'équation de droite idéale étant : $\text{Sortie} = 1 * \text{Cible} + 0$

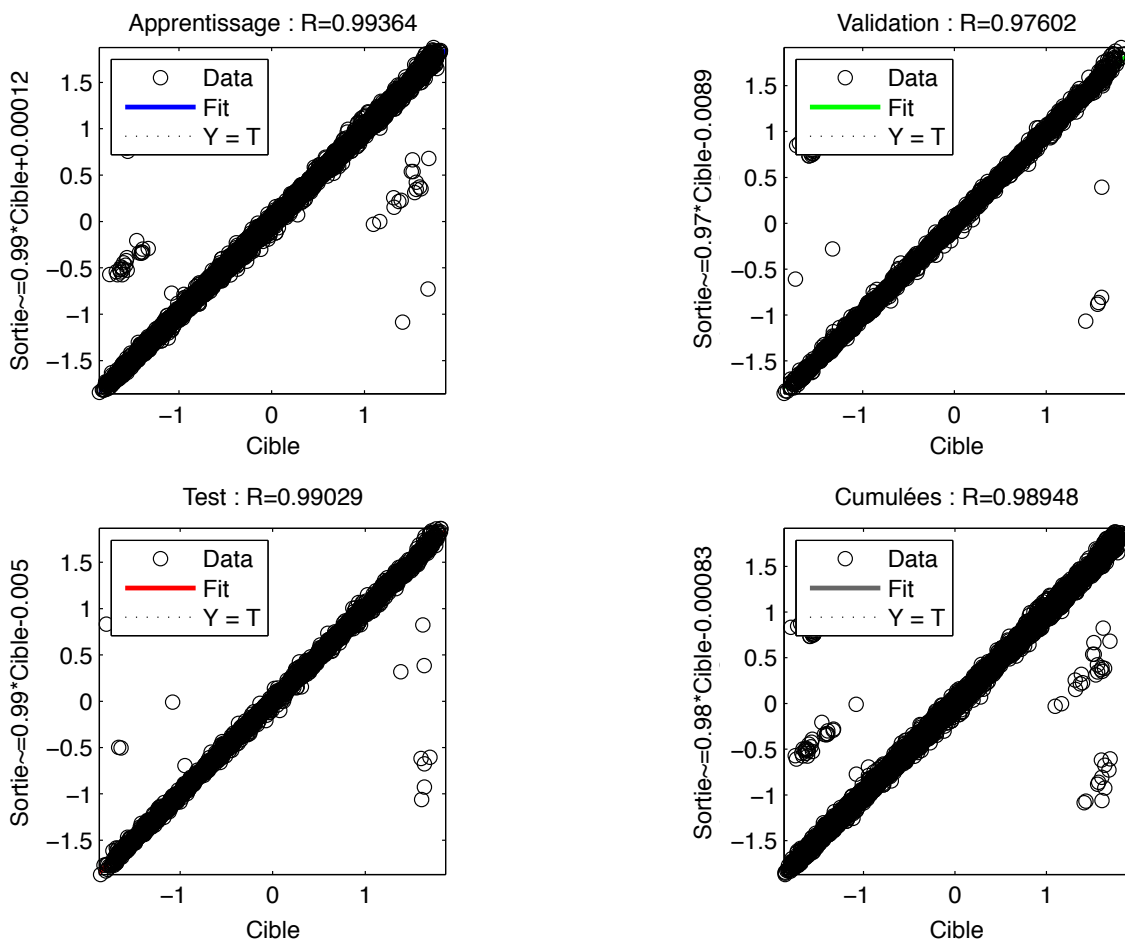


Figure 113: Régression en fonction des bases de données utilisées.

Le coefficient de corrélation linéaire : R est aussi indiqué pour chacune des bases de données. Il indique la corrélation entre les données en sortie du réseau de neurones (les positions du SPIDAR calibré) et les données ciblées (les positions du tracking optique). Il désigne aussi le coefficient directeur de la droite de régression linéaire. Ce coefficient est compris entre -1 et 1 . Plus sa valeur absolue tend vers 1 et plus les données sont corrélées linéairement. Comme le montrent ces résultats, le réseau de neurones a permis de déterminer une fonction de régression suffisamment efficace pour que la quasi-totalité des données en sortie du réseau de neurones soient corrélées avec les données cibles. Ainsi, on obtient un coefficient de corrélation linéaire supérieur à 0.99 pour les données en apprentissage (60% des données totales). Pour l'ensemble des données cumulées (100% des données), le coefficient de corrélation linéaire reste très élevé (0.98).

Le graphique 114 représente l'erreur absolue sur la position du SPIDAR avant et après calibration. Comme nous pouvons l'observer, avant calibration, l'erreur absolue sur la position donnée par le SPIDAR augmente en fonction de la distance du point d'initialisation. Cependant, cette erreur peut également être faible même en étant situé loin du point d'initialisation. C'est pour cette raison que le graphique de gauche ne montre pas une courbe, mais plutôt une surface triangulaire, témoignant de l'aspect non linéaire de la répartition de l'erreur absolue dans l'espace. Nous aboutissons donc bien à la même conclusion que pour le graphique 104. Le graphique de droite démontre l'utilité du réseau de neurones. En effet, on peut très bien observer que l'erreur absolue sur la position par rapport au point d'initialisation est répartie de manière beaucoup plus homogène après calibration, confirmant que le réseau de neurones a permis de diminuer fortement l'erreur sur la position, augmentant ainsi sa précision.

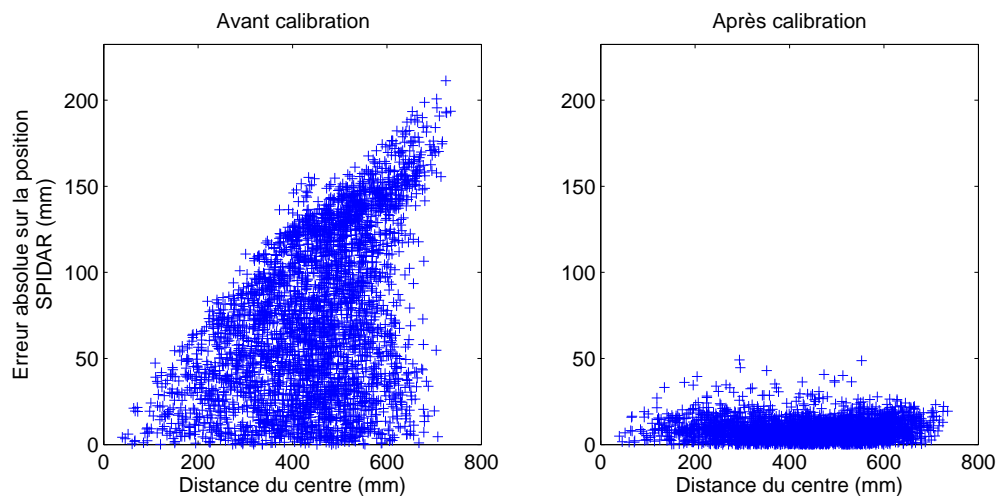


Figure 114: Erreurs absolues sur la position du SPIDAR par rapport à la distance à laquelle la mesure a été prise du point d'initialisation, avant et après calibration par le réseau de neurones.

5.5.5.1 Distribution de l'erreur absolue dans l'espace de travail

Cette constatation est confirmée par le graphique 115. Nous pouvons distinguer que l'apprentissage du réseau de neurones a grandement amélioré la précision sur la position du SPIDAR. A la vue du second graphique qui dispose d'une plage d'erreurs diminuée

de moitié par rapport au premier et donc d'une gamme de couleur différente, on peut clairement affirmer que les erreurs résiduelles ne sont pas du bruit de mesure. En effet, dans le cas d'un bruit de mesure, celui-ci serait réparti de manière homogène dans l'espace. Afin de nous en assurer, nous avons tracé l'histogramme des erreurs absolues sur la position du SPIDAR par rapport à celle du tracking optique afin d'étudier en détail l'étendue des erreurs.

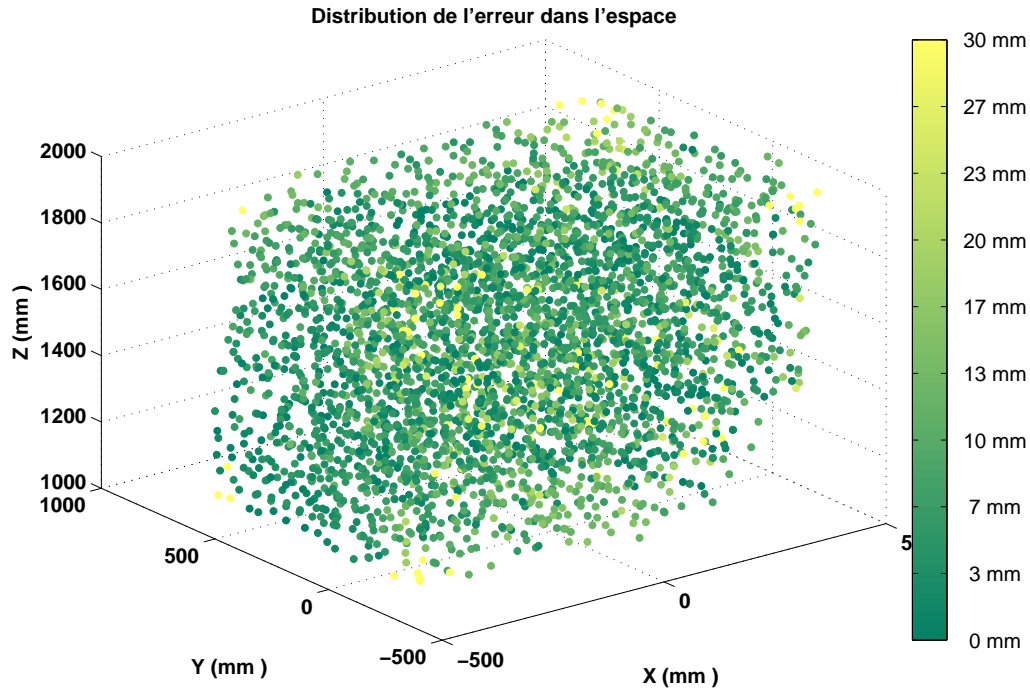


Figure 115: Représentation de l'erreur absolue dans l'espace sur la position, avec une plage d'erreur réduite de moitié.

5.5.5.2 Histogramme des erreurs

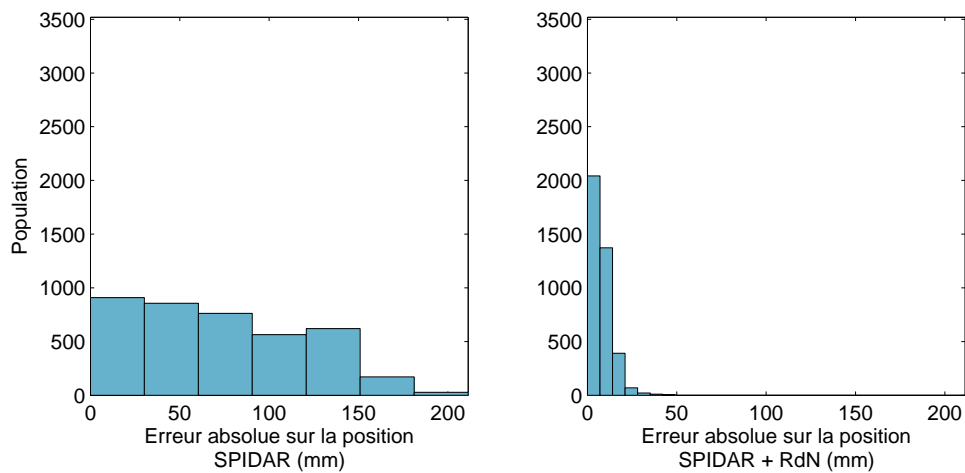


Figure 116: Erreur absolue sur la position du SPIDAR avant et après calibration par le réseau de neurones.

ERREUR ABSOLUE	BRUT	RDN
MOYENNE (MM)	72.86	7.80
ECART-TYPE (MM)	47.06	5.84
MAX (MM)	155.19	49.19

Table 4: Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par un réseau de neurones.

La figure 116 et le tableau 4 représentent respectivement l'histogramme des erreurs absolues sur la position du SPIDAR avant (brut) et après calibration par le réseau de neurones (RDN), ainsi que les valeurs caractéristiques de ces erreurs (moyenne, écart-type et valeur maximale). Un histogramme étalé signifie que l'erreur absolue varie et qu'on a une mauvaise précision sur la position du SPIDAR : c'est le cas pour le SPIDAR avant calibration. *A contrario*, un histogramme qui est réduit vers le côté gauche (c'est-à-dire que les erreurs absolues sont très petites) démontre une bonne précision sur la position du SPIDAR : c'est le cas après calibration par le réseau de neurones. Ces résultats sont confirmés par l'observation des valeurs caractéristiques des erreurs absolues. Comme nous pouvons le voir, la moyenne des erreurs absolues est passée d'environ 72 mm pour le SPIDAR avant calibration (BRUT dans le tableau) à 7.8 mm pour le SPIDAR calibré par le réseau de neurones (RDN dans le tableau). L'écart-type, témoignant de la dispersion des erreurs absolues, quant à lui, été divisé par 9, passant d'environ 47 mm à 5.8 mm pour le SPIDAR calibré.

5.5.6 Résultats du réseau de neurones en généralisation

Afin d'évaluer les performances d'un réseau de neurones, il ne suffit pas juste de le tester avec les données qui ont servi à l'apprentissage, il faut également étudier son comportement face à des données auxquelles il n'a jamais été confronté. On appelle cette opération la généralisation.

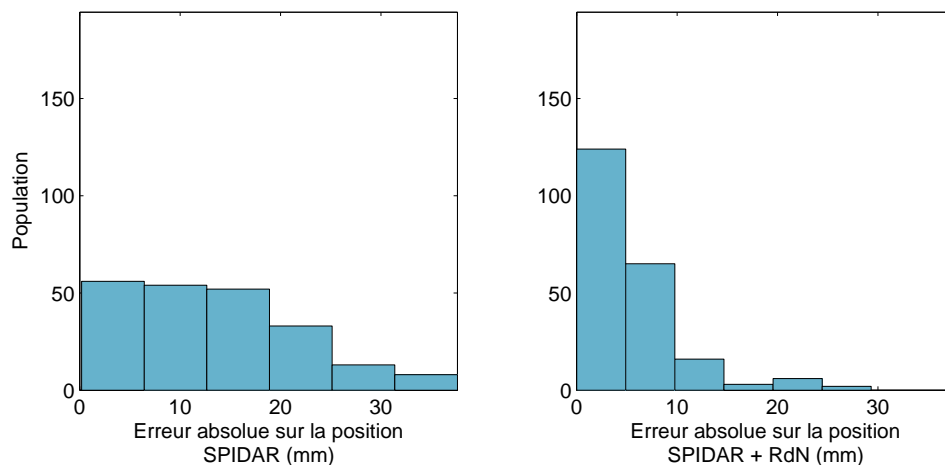


Figure 117: Erreur absolue moyenne sur la position du SPIDAR avant et après le réseau de neurones sur la base de données 1.

ERREUR ABSOLUE	BRUT	RDN
MOYENNE (MM)	13.23	5.19
ECART-TYPE (MM)	8.41	5.13
MAX (MM)	37.60	22.79

Table 5: Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par le réseau de neurones sur la base de données 1.

Ces données sont récupérées lors d'une campagne de mesure, que nous appelons base de données en généralisation, durant laquelle on effectue des trajectoires aléatoires avec l'effecteur du SPIDAR, en enregistrant à la fois les positions données par le SPIDAR et celles données par le tracking optique ART. Nous avons créé deux bases de données afin de tester les performances du réseau de neurones en généralisation.

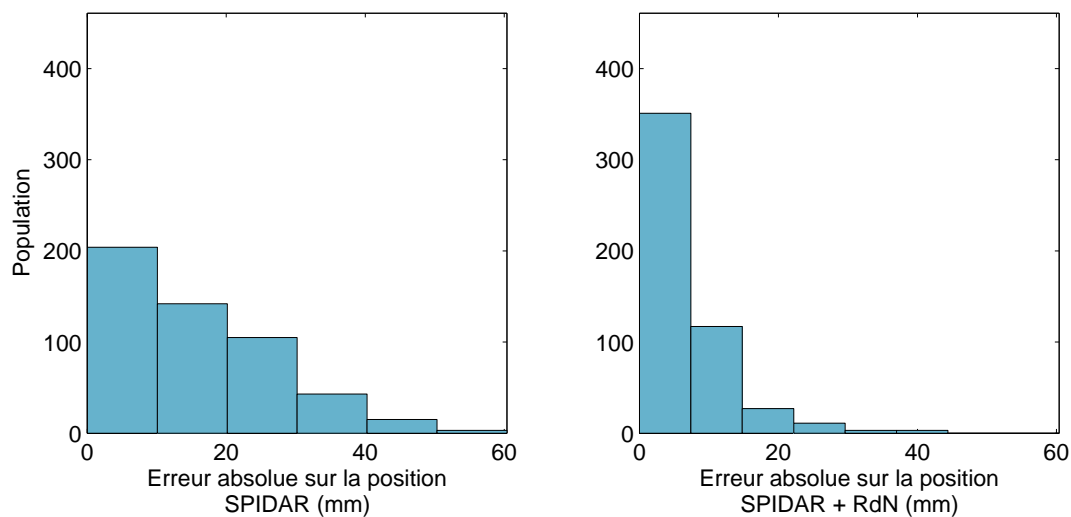


Figure 118: Erreur absolue moyenne sur la position du SPIDAR avant et après le réseau de neurones sur la base de données 2.

ERREUR ABSOLUE	BRUT	RDN
MOYENNE (MM)	15.79	6.59
ECART-TYPE (MM)	11.64	6.41
MAX (MM)	54.53	44.39

Table 6: Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par le réseau de neurones sur la base de données 2.

On peut remarquer que le réseau de neurones conserve de bonnes performances avec des données en généralisation. Ainsi l'erreur absolue moyenne a été diminuée par 2.5 fois, passant ainsi de 13.23 mm à 5.19 mm. Tout comme pour la première base de données, le réseau de neurones a pour ces données, conservé de très bonnes performances

en permettant de diminuer de moitié l'erreur absolue moyenne et en divisant par 2 l'écart-type par rapport au SPIDAR non calibré.

REMARQUE: La dispersion des erreurs absolues, représentée par la valeur de l'écart-type, est ici moindre par rapport à celle lors de l'apprentissage car la base de donnée utilisée couvre un espace de travail moindre.

5.5.7 Conclusion sur l'utilisation d'un réseau de neurones

A partir des différents résultats présentés, nous pouvons conclure que le réseau de neurones utilisé est robuste à toutes les données présentées, dès lors qu'elles appartiennent à l'espace de travail du SPIDAR. Un tel résultat n'est pas surprenant, puisque le protocole d'apprentissage a été très rigoureux. De plus, le nombre de données et la répartition de celles-ci dans l'espace permettent un apprentissage qui soit à même de couvrir tout l'espace de travail du SPIDAR, garantissant ainsi au réseau de neurones la possibilité d'effectuer une très bonne interpolation. Cependant, l'inconvénient des réseaux de neurones est que nous ne pouvons pas être certains que nous avons atteint le minimum local et par conséquent, nous ne pouvons pas garantir que nos résultats soient optimaux. C'est pour cette raison que dans la partie suivante, nous avons choisi une approche par les SVRs.

5.6 CALIBRATION DU SPIDAR PAR LES SVRS

5.6.1 Présentation

Afin de pouvoir vérifier les performances de notre réseau de neurones, ainsi que pour pouvoir réaliser une étude comparative de la meilleure solution à adopter, nous avons utilisé les SVRs qui sont l'application des SVMs au problème de la régression. Pour une présentation de cette méthode, le lecteur peut consulter la partie [3.4.2.4](#).

5.6.2 Choix des paramètres pour les SVRs

Après avoir testé plusieurs ensembles de paramètres nous avons déterminé de manière empirique que c'est le couple de paramètres $\{\epsilon, C\}$ qui offre les meilleures performances était :

$$\{\epsilon = 0.1, C = 1\}$$

5.6.3 Performances des SVRs sur l'apprentissage

Comme pour les précédentes méthodes, nous représentons la variation de l'erreur absolue en fonction de la position de l'effecteur par rapport au point d'initialisation avant et après calibration (voir figure [119](#)). Comme nous pouvons l'observer, la calibration de la position du SPIDAR par les SVRs est efficace. Nous constatons, une nette diminution

des erreurs absolues sur la position. Ces résultats semblent similaires à ceux obtenus avec le réseau de neurones (figure 114), il faut néanmoins noter une plus grande dispersion statistique avec cette approche qu'avec le réseau de neurones.

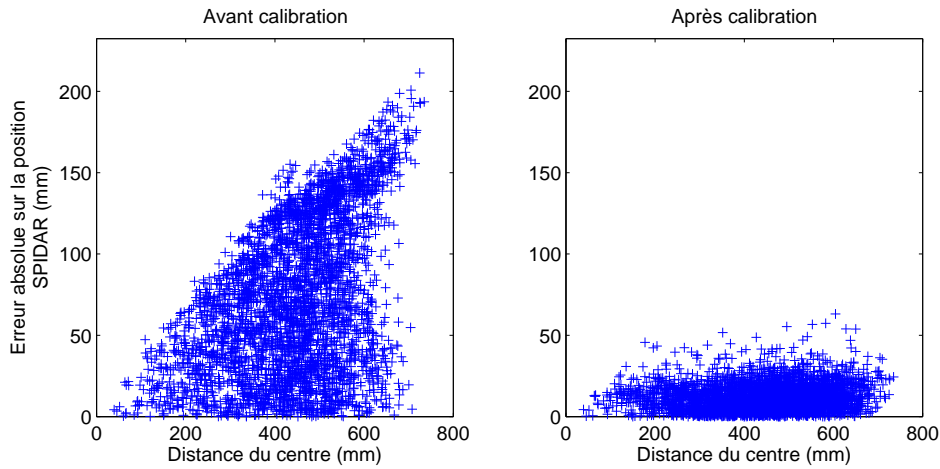


Figure 119: Erreurs absolues sur la position du SPIDAR par rapport à la distance à laquelle la mesure a été prise du point d'initialisation, avant et après calibration par les SVRs.

5.6.3.1 Distribution de l'erreur absolue dans l'espace de travail

La figure 120 confirme l'efficacité de la calibration par les SVRs. Nous remarquons que la majorité des erreurs absolues supérieures à 30 mm sont situées aux limites de l'espace de travail, alors qu'à l'intérieur de celui-ci, l'erreur est inférieure à 20 mm. Nous faisons le même constat que pour la figure précédente. On remarque également que ce graphique est similaire à celui obtenu après calibration par le réseau de neurones.

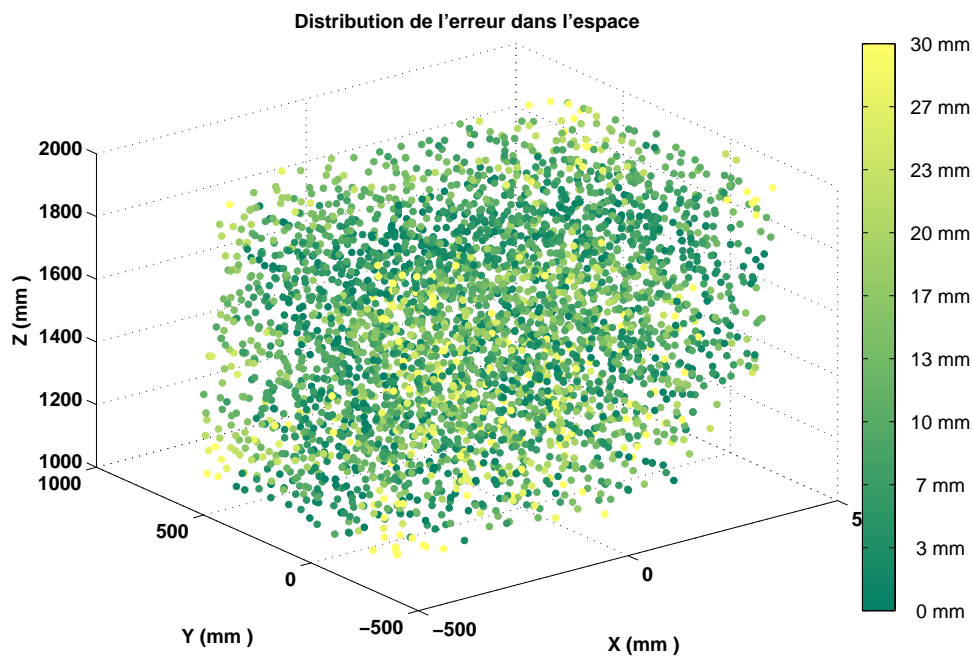


Figure 120: Répartition de l'erreur absolue sur la position du SPIDAR dans l'espace de travail.

5.6.3.2 *Histogramme des erreurs*

La figure 121 et le tableau 7 démontrent l'efficacité de la calibration par les SVRs. On diminue ainsi l'erreur absolue moyenne sur la position de 72.86 mm à 11.48 mm, on a donc une nette augmentation de la précision moyenne. L'écart-type a également très fortement diminué passant ainsi de 47.06 mm à 8.14 mm, ce qui témoigne d'une forte diminution de la dispersion statistique.

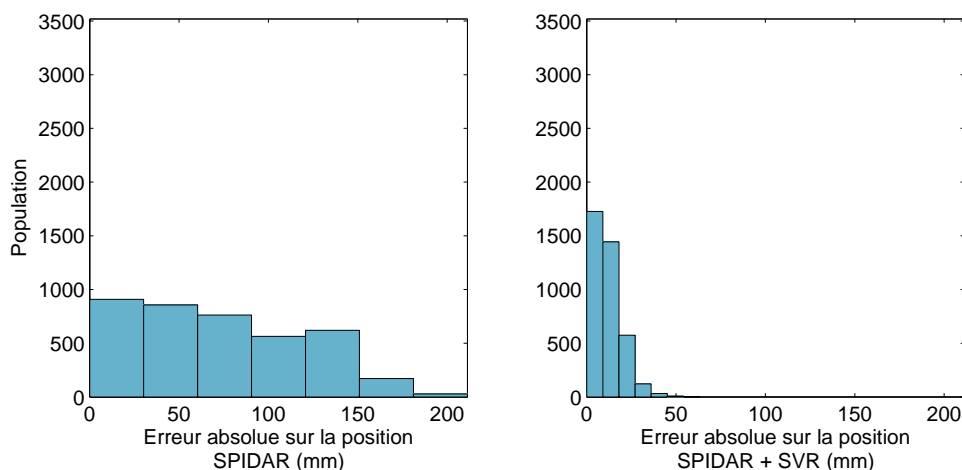


Figure 121: Erreur absolue sur la position du SPIDAR avant et après calibration par les SVRs.

ERREUR ABSOLUE	BRUT	SVR
MOYENNE (MM)	72.86	11.48
ECART-TYPE (MM)	47.06	8.14
MAX (MM)	155.19	42.19

Table 7: Valeurs caractéristiques des erreurs absolues sur la précision du SPIDAR obtenues avec une calibration par les SVRs.

5.7 MÉTHODE DE CALIBRATION RETENUE ET APPLICATION PRATIQUE DE LA CALIBRATION

Il nous faut maintenant faire un choix sur la meilleure approche à adopter pour la calibration du SPIDAR. La figure 122 et le tableau 8 synthétisent les performances des différentes approches que nous avons testées précédemment.

Comme nous pouvons le constater, c'est la calibration par le réseau de neurones qui offre les meilleures performances, avec une erreur absolue moyenne de 7.80 mm, en seconde place se situe la calibration par les SVRs avec 11.48 mm d'erreur moyenne. La calibration par régression linéaire (IP₁) est la méthode offrant les plus mauvaises performances avec une erreur moyenne de plus 33 mm. En terme, de dispersion statistique, le classement des méthodes de calibration est le même. Ainsi, le réseau de neurones est la méthode souffrant le moins de dispersion statistique avec une valeur en dessous

de 6 mm, puis vient l'approche par les SVRs avec une dispersion d'environ 8 mm et en dernier la régression linéaire avec 16 mm. On peut donc conclure que la meilleure approche parmi celles évaluées, pour calibrer la position du SPIDAR et ainsi améliorer sa précision, est la calibration par le réseau de neurones.

5.7.1 Tableau comparatif des différentes méthodes de correction

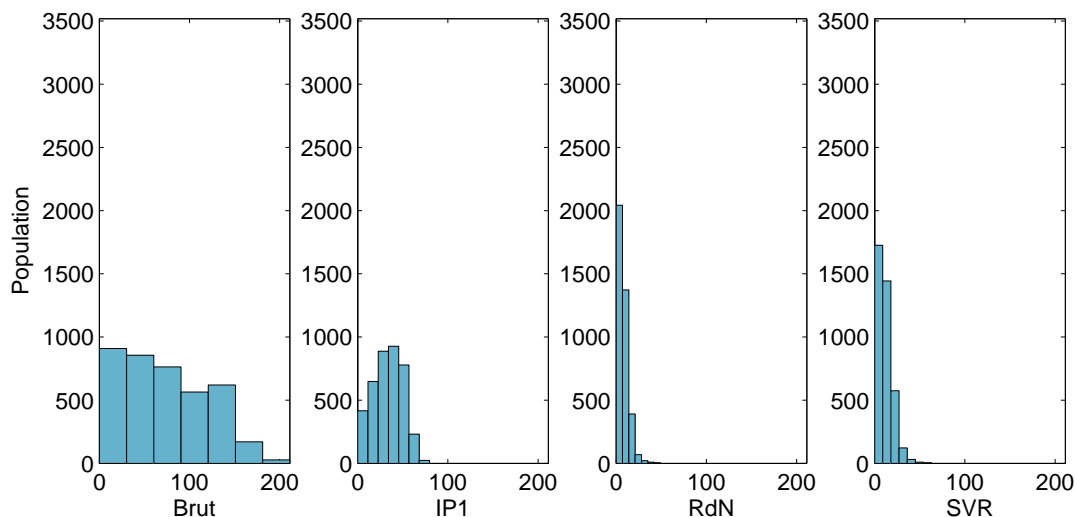


Figure 122: Etude comparative des erreurs sur la calibration en fonction des différentes méthodes.

ERREUR ABSOLUE	BRUT	IP1	RDN	SVR
MOYENNE (MM)	72.86	33.53	7.80	11.48
ECART-TYPE (MM)	47.06	16.01	5.84	8.14
MAX (MM)	155.19	79.51	49.19	42.19

Table 8: Comparaison des valeurs caractéristiques des erreurs absolues.

5.7.2 Application pratique

Une fois la meilleure approche choisie, nous devons concevoir un système de tracking hybride qui est une combinaison du tracking provenant du SPIDAR et du système optique ART. En observant la disponibilité de l'un ou l'autre des systèmes de tracking, on prend la décision de basculer vers le système le plus fiable. Comme nous l'avons dit précédemment, le tracking le plus fiable en termes de précision est celui fourni par la technologie optique, c'est donc celui-ci doit être privilégié en priorité. Dans le cas où il est indisponible, on utilise le suivi du SPIDAR. Cette démarche est comparable à celle effectuée par [Birkfellner et al](#) sur un tracking hybride basé sur les technologies ultrasons et magnétique [[Birkfellner et al, 1998](#)].

5.7.3 Logigramme de basculement automatique:

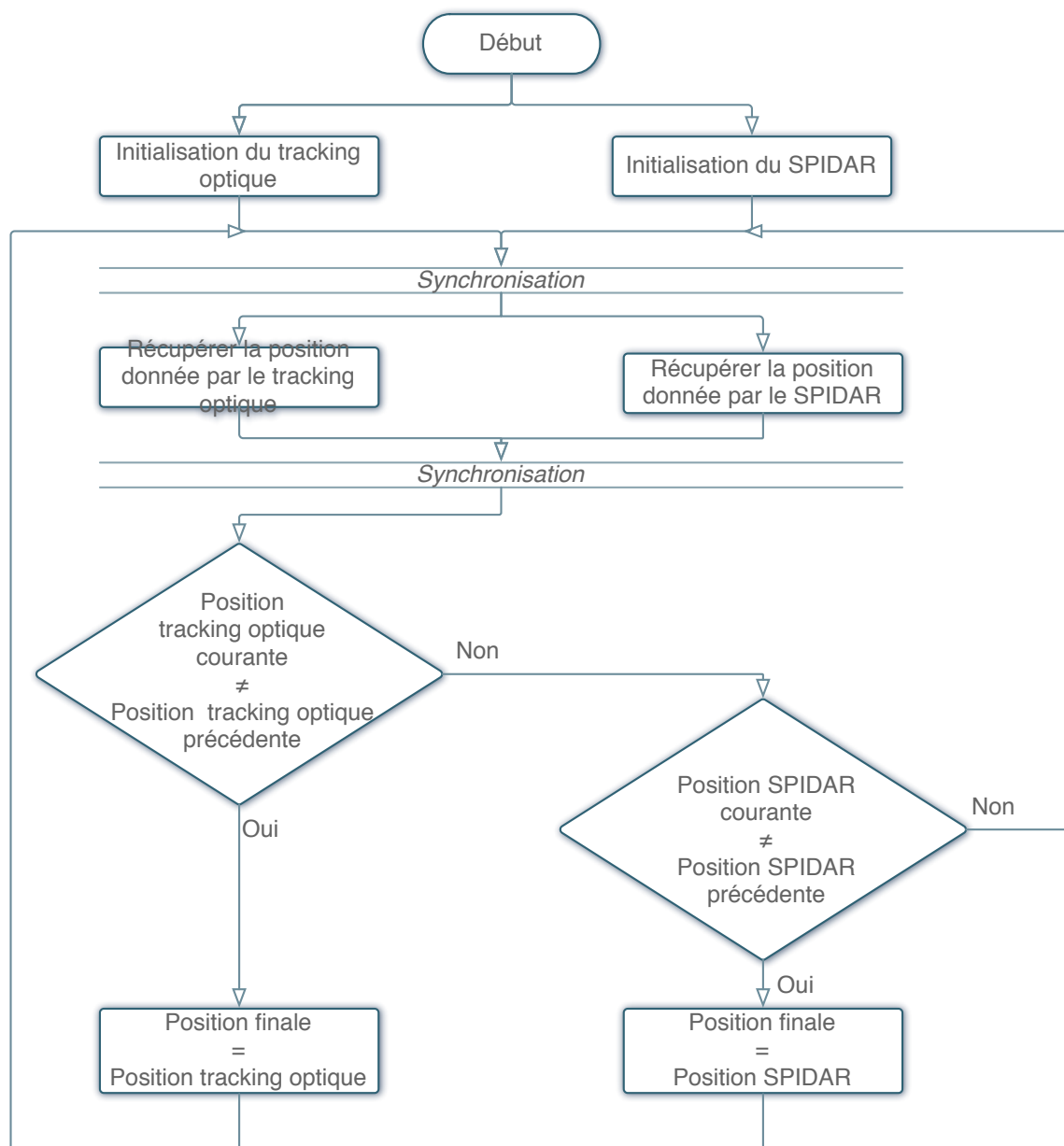


Figure 123: Algorithme de basculement entre les deux systèmes.

Le basculement automatique entre les deux systèmes se réalise de la manière suivante :

1. Synchronisation et récupération des données provenant du SPIDAR et du système de tracking optique;
2. Calibration du SPIDAR par le réseau de neurones;
3. Basculement entre le système ART et le SPIDAR estimé en fonction de l'activité du tracking optique.

Le logigramme représenté en figure 123 résume le mode de fonctionnement du basculement automatique.

5.7.4 Résultats:

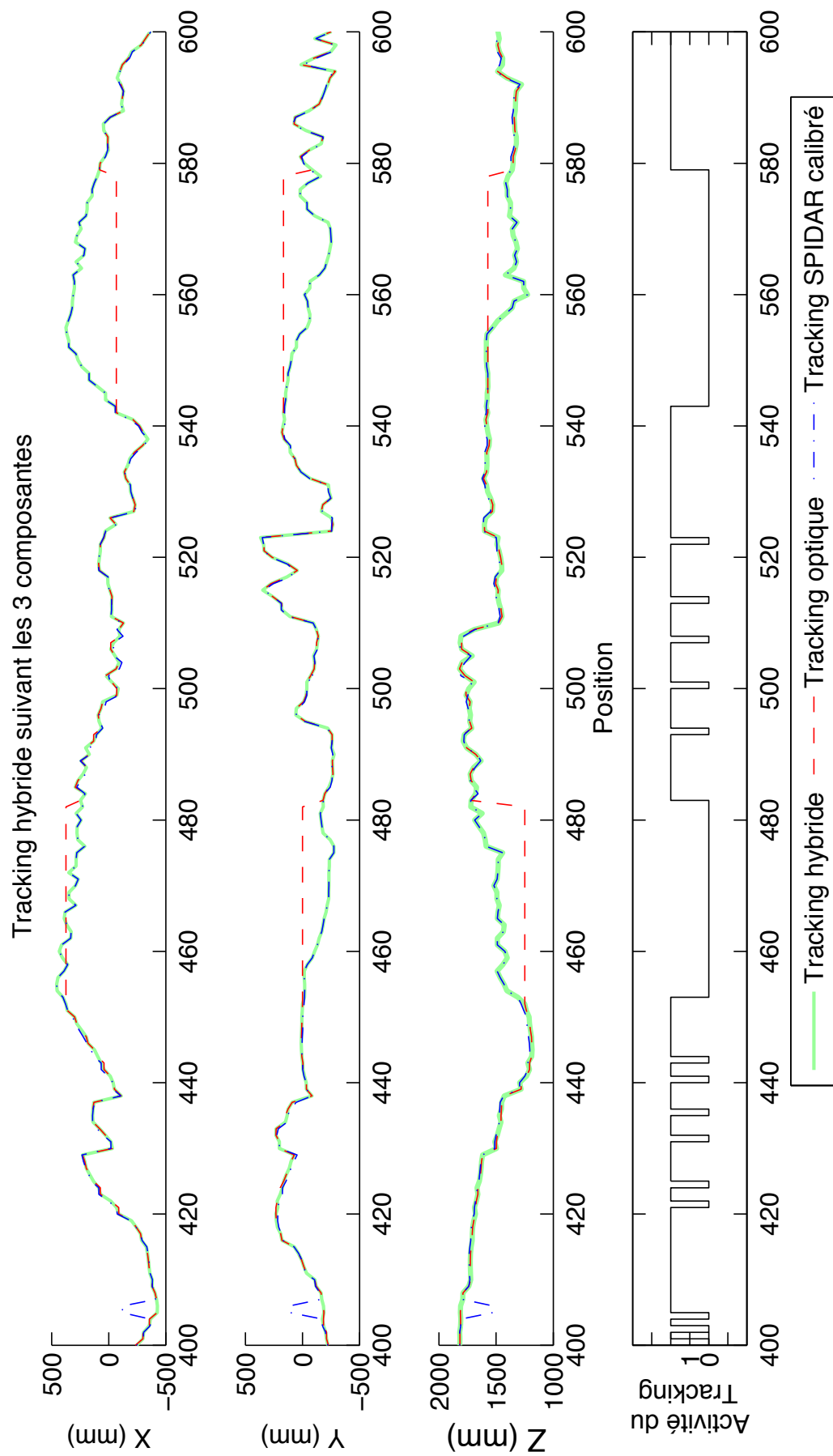


Figure 124: Exemple d'une trame du tracking hybride mis en place

Comme le montre la figure 124, le tracking hybride que nous proposons répond à nos attentes. Le tracking fourni par le SPIDAR est suppléé par le tracking optique lorsque celui-ci est inopérant (activité du tracking égale à zéro), comme par exemple dans le cas d'une occultation. Ce phénomène est bien représenté entre les trames 450 à 480 et 540 à 580. On remarque aussi, que dans le cas où la calibration du tracking n'est pas très efficace, ce que l'on peut voir à la trame 408, le tracking optique étant prioritaire, l'écart du tracking du SPIDAR calibré n'est pas pris en compte dans le tracking hybride.

5.8 CONCLUSION

Dans ce chapitre, nous avons proposé différentes approches pour calibrer au mieux le SPIDAR par rapport à une référence fournie par un système de tracking optique. Nous avons proposé un système d'initialisation multimodale pour le SPIDAR, permettant de guider l'utilisateur dans l'accomplissement d'une initialisation avec une précision de l'ordre du demi-millimètre.

Nous avons par la suite, procédé à la calibration du SPIDAR, par rapport au système du tracking optique, à l'aide de trois méthodes : la régression linéaire, les réseaux de neurones et les SVRs. Parmi toutes ces méthodes de calibration, c'est le réseau de neurones qui s'avère le plus efficace. Il a démontré son aptitude à estimer des données en sortie d'un système, sans avoir de connaissance *a priori* sur le modèle mathématique utilisé. Il faut néanmoins noter, que les résultats obtenus avec les SVRs sont acceptables. La régression linéaire est, quant à elle, inefficace pour le système non linéaire du SPIDAR car elle n'opère que dans les zones où le système est quasi-linéaire.

Cette calibration nous a ensuite permis de concevoir un tracking hybride basé sur le SPIDAR et le système de tracking optique. Avec cette solution, on utilise en priorité le tracking optique et lorsqu'il est défaillant, on bascule vers le tracking du SPIDAR calibré. La calibration de celui-ci permet de basculer d'un système à l'autre de manière transparente. Nous avons donc un suivi de la position de la main de l'utilisateur qui est continu, garantissant donc un bon fonctionnement de la technique d'interaction 3D Fly-Over et une interaction 3D continue.

EVALUATION DE L'UNITÉ D'INTERACTION 3D ADAPTATIVE

6.1 SCÉNARIO D'ÉVALUATION MIS EN PLACE

Evaluer Fly-Over est difficile puisque c'est une technique d'interaction 3D qui est conçue pour réaliser toutes les tâches de l'interaction 3D sans discontinuité. Il est par conséquent très difficile de la comparer avec d'autres techniques qui elles sont conçues pour une seule voire deux tâches au maximum. D'un autre côté, il serait dommage d'évaluer Fly-Over sans préserver la continuité qu'il procure. C'est pour cette raison que nous n'évaluerons la technique d'interaction 3D elle-même, mais seulement sa composante contrôle d'application c'est-à-dire l'unité d'interaction 3D adaptative (présentée dans la section 4.5).

Nous avons donc fait le choix d'établir un scénario dans lequel l'utilisateur est amené à interagir en continu afin d'évaluer l'efficacité du système de basculement automatique de la tâche de l'interaction 3D face à un basculement manuel. Pour isoler au mieux les paramètres qui nous permettront d'évaluer l'efficacité du système de basculement de tâche automatique nous désactiverons tout paramètre extérieur à même d'influer sur les résultats de cette évaluation. Ainsi, le système de reconnaissance de geste, l'affichage de la part des transformations géométriques appliquées et l'énonciation des tâches de l'interaction 3D seront désactivés.

Cette évaluation se déroulera en deux phases : une phase d'évaluation objective, basée sur l'observation et l'interprétation de certains paramètres recueillis durant les évaluations, et une phase d'évaluation subjective sous la forme d'un questionnaire que l'on soumettra aux personnes ayant participé aux évaluations.

6.2 PROFILS DES PARTICIPANTS À L'ÉVALUATION

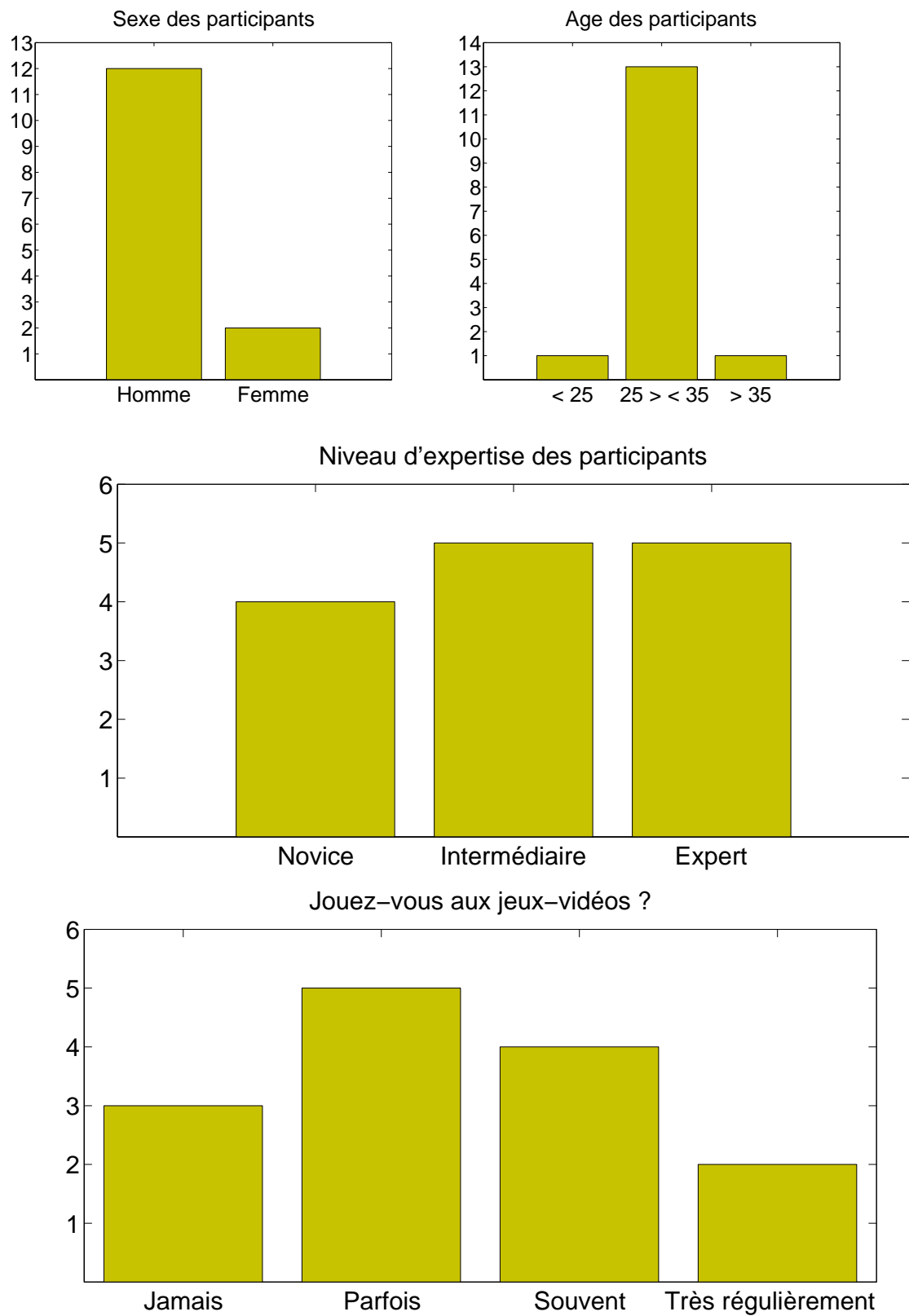


Figure 125: Informations sur le profil des participants 1/2

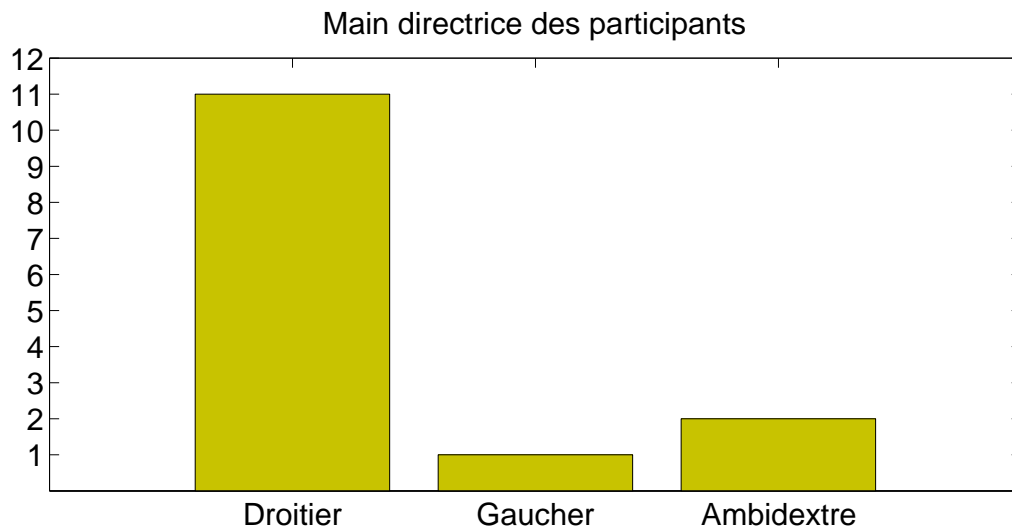


Figure 126: Informations sur le profil des participants 2/2

Comme les figures 125 et 126 peuvent le montrer, le panel des participants est assez bien distribué au niveau de l'expertise en réalité virtuelle ou bien au niveau de leur expérience dans les jeux-vidéo. Bien évidemment, parmi les participants, la majorité étaient des droitiers, le reste étant des gauchers et les ambidextres, qui notre cas, étaient des gauchers ayant utilisé leur main droite pour les évaluations.

6.3 EVALUATION OBJECTIVE

6.3.1 Déroutement de l'évaluation objective

L'évaluation consiste à faire réaliser à l'utilisateur une succession de tâches de l'interaction 3D en utilisant le système de tracking hybride (optique/SPIDAR) avec le système de basculement automatique des tâches de l'interaction 3D et avec le système manuel.

Dans un premier temps, l'utilisateur devra se familiariser avec le système dans sa globalité dans une phase d'apprentissage. Il pourra ainsi, prendre ses repères et comprendre comment utiliser le système de tracking, la technique d'interaction 3D Fly-Over ainsi que le système automatique et manuel de changement des tâches.

Une fois cet apprentissage effectué, on commence l'évaluation à proprement parler. L'utilisateur devra accomplir 20 fois la même tâche, qui consistera en :

1. un déplacement jusqu'à un objet;
2. la sélection puis la saisie de cet objet;
3. le déplacement de cet objet jusqu'à un autre emplacement;
4. le dépôt de celui-ci dans une certaine zone.

Sur ces 20 essais, 10 seront effectués avec le mode de basculement automatique et les 10 autres se feront en mode manuel. Les essais avec chacun des systèmes (auto et manuel)

s'enchaîneront en alternance afin d'éviter le phénomène d'apprentissage favorisé pour l'un ou pour l'autre.

En mode manuel, l'utilisateur se verra attribuer un flystick dans la main non directrice (la main gauche pour un droitier, la main droite pour un gaucher) afin de pouvoir passer d'une tâche à l'autre. La configuration des boutons sur le flystick est la suivante:

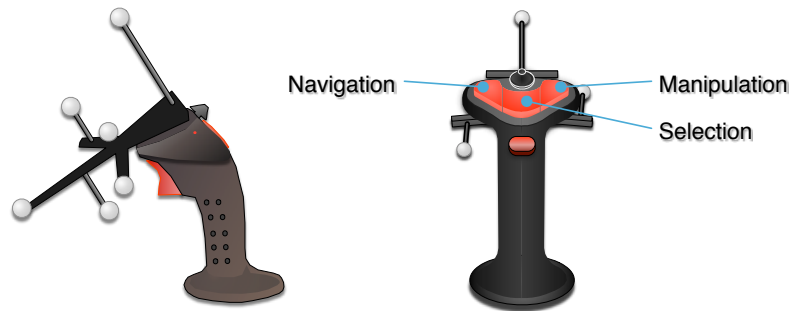


Figure 127: Configuration des boutons du flystick pour la seconde phase de l'évaluation.

Voici le déroulement détaillé d'un essai de l'évaluation objective :

1. Un affichage textuel de la consigne est affiché à l'utilisateur afin de lui rappeler les objectifs de l'évaluation.
2. La première partie de cette évaluation concerne la navigation directe de point en point assez distant les uns des autres. L'utilisateur commence à un point A, il observe de ce point de départ une zone illuminée en vert, qui correspond au prochain point de contrôle (B) auquel il doit se rendre. Les paramètres observés sont :
 - Le temps mis entre chaque point de contrôle;
 - Le temps global mis pour aller du premier point au dernier point de contrôle.

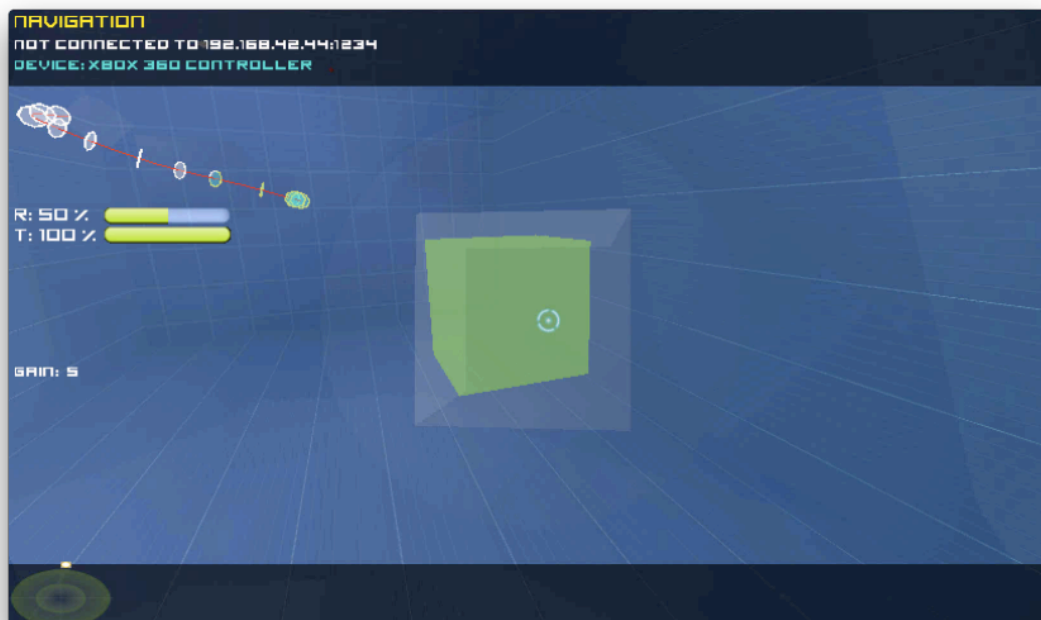


Figure 128: La phase de navigation directe de point en point avec Fly-Over.

3. La 3^{ème} étape consiste à sélectionner puis manipuler une sphère jaune. Non seulement l'utilisateur doit déplacer cet objet de sa position initiale à une position désigné par un objet *fantôme*, mais il devra le positionner avec le plus de précision possible. Les paramètres observés sont :
 - Le temps mis pour sélectionner l'objet;
 - Le temps global mis pour réaliser toutes les tâches de sélection.
4. Dans la quatrième et dernière étape, l'utilisateur doit manipuler un objet d'un point A vers un point B. Les paramètres observés sont :
 - Le temps mis pour réaliser le déplacement du point A au point B;
 - La précision sur l'orientation de l'objet par rapport à celle demandée.

6.3.2 Paramètres observés pour l'évaluation

De manière globale, les deux principaux paramètres que nous allons observer sont :

- l'ordre dans lequel les 3 tâches de l'interaction 3D (navigation, sélection et manipulation) se sont enchainées;
- le temps passé sur chacune de ses tâches.

Le scénario d'évaluation élaboré est tel que nous connaissons l'ordre d'enchainement des tâches optimal pour terminer le scénario. Cet enchainement est le suivant:

Sélection $\xrightarrow{1}$ Navigation $\xrightarrow{2}$ Sélection $\xrightarrow{3}$ Manipulation $\xrightarrow{4}$ Navigation $\xrightarrow{5}$ Manipulation $\xrightarrow{6}$ Sélection.

Avec :

1. Sélection (tâche par défaut) vers la navigation,
2. Navigation jusqu'à la sphère rouge,
3. Sélection de la sphère rouge puis déplacement de celle-ci,
4. Navigation avec la sphère rouge,
5. Placement précis de la sphère rouge dans la zone de dépôt,
6. Relâchement de l'objet dans la zone de dépôt.

6.3.3 Résultats de l'évaluation objective et interprétation

6.3.3.1 Nombre de basculement de tâches de l'interaction 3D

A - ANALYSE POUR TOUS LES PARTICIPANTS

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	9.28	0.90	0.95	8.07	10.64
MANUEL	7.35	0.07	0.26	7.00	7.79

Table 9: Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Tous les participants.

Le tableau 9 nous montre le nombre de basculements moyen de la tâche de l'interaction 3D, réalisés par l'ensemble des participants, effectués au fur et à mesure des 10 essais. Comme nous pouvons le constater, avec le mode automatique (avec unité d'interaction 3D adaptative), la tâche s'est réalisée en moyenne avec un basculement supplémentaire par rapport au mode manuel (sans unité d'interaction 3D adaptative), qui lui est très proche du nombre minimal de basculement à effectuer (7.2 par rapport à un minimum de 7). La variance témoigne que l'unité d'interaction 3D adaptative est plus susceptible de se tromper sur le basculement d'une tâche que l'utilisateur. Un tel résultat était bien évidemment à prévoir, puisqu'en mode manuel, c'est l'utilisateur qui contrôle ce basculement et que bien évidemment il est généralement d'accord avec ses propres choix.

B - ANALYSE POUR LE GROUPE DES EXPERTS

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	7.42	0.23	0.48	7.00	8.00
MANUEL	7.12	0.10	0.32	7.00	8.00

Table 10: Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Groupe des experts.

Comme le montre le tableau 10, nous constatons que les participants experts n'ont aucun problème à utiliser l'un ou l'autre des systèmes de basculement de la tâche de l'interaction 3D. En effet, nous avons un nombre moyen d'environ 7 basculements avec une variance faible pour les deux systèmes. Il faut remarquer que sur l'ensemble des participants, le maximum de basculement atteint est de 8 avec l'unité d'interaction 3D adaptative.

C - ANALYSE POUR LE GROUPE DES INTERMEDIAIRES

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	10.34	4.63	2.15	7.20	13.40
MANUEL	7.52	0.26	0.51	7.00	8.20

Table 11: Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Groupe des intermédiaires.

Le tableau 10 démontre, que pour le groupe des intermédiaires, le constat n'est pas le même. L'unité d'interaction 3D adaptative se trompe un peu plus avec ce groupe de personnes avec un nombre moyen de 10 basculements de tâches de l'interaction 3D. Cependant, ce nombre de basculement n'est pas important car, finalement il ne représente qu'une seule erreur de l'unité d'interaction 3D adaptative. En effet, pour une erreur de tâche de l'interaction 3D, il y aura deux basculements : un pour le basculement vers la mauvaise tâche et un autre pour revenir à la bonne tâche.

D - ANALYSE POUR LE GROUPE DES NOVICES

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	10.28	2.81	1.68	8.00	12.50
MANUEL	7.42	0.20	0.44	6.75	8.25

Table 12: Valeurs caractéristiques du nombre de basculements de la tâche de l'interaction 3D - Groupe des novices.

Pour le groupe des novices, même constat que pour le groupe des intermédiaires avec également un nombre moyen de 10 basculements.

6.3.3.2 Temps de réalisation de l'épreuve

A - ANALYSE POUR TOUS LES PARTICIPANTS

La figure 129 représente le temps moyen de réalisation de l'épreuve imposé par l'évaluation en fonction des essais. Ces temps sont basés sur les temps de tous les participants. On distingue très nettement une courbe d'apprentissage aussi bien pour le mode automatique que pour le mode manuel.

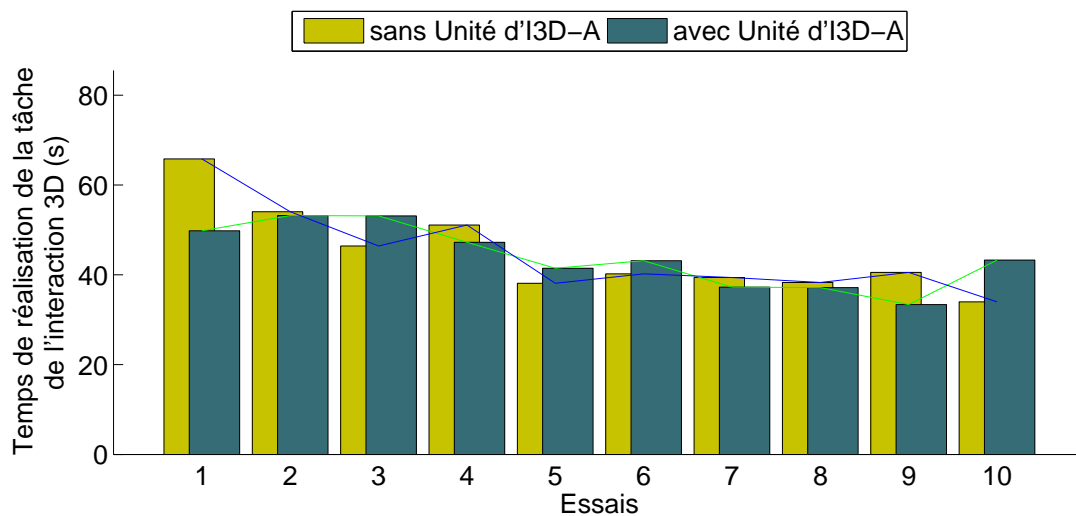


Figure 129: Temps moyens de la réalisation de l'épreuve demandée en fonction des essais.

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	43.90	47.19	6.87	33.37	53.19
MANUEL	44.79	93.47	9.67	33.98	65.81

Table 13: Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Tous les participants.

Mais ce sont les valeurs caractéristiques du tableau 13 qui sont les plus intéressantes. En effet, elle démontre que l'unité d'interaction 3D adaptative ne pénalise pas le temps de réalisation de l'épreuve mais qu'elle ne l'améliore pas non plus. Un tel résultat paraît logique dans la mesure où l'on considère que le système de basculement ne peut pas être plus efficace que la décision humaine. En revanche ce résultat démontre que l'unité d'interaction 3D adaptative peut se substituer à la décision humaine et ainsi soulager cognitivement l'utilisateur qui pourra ainsi maintenir son attention gagnée sur un autre problème. De plus, si l'on regarde attentivement la valeur de la variance pour le mode automatique, on peut remarquer qu'elle est deux fois moins élevée que celle en mode manuel. D'un point de vue expérimental, cela signifie que les utilisateurs obtiennent des temps plus réguliers avec le mode automatique qu'en mode manuel. Ceci s'explique par le fait que l'unité d'interaction 3D adaptative réagit de la même manière quel que soit l'utilisateur prenant ainsi la décision au moment jugé comme optimal. Au contraire de la décision humaine, qui elle varie d'un utilisateur à un autre.

B - ANALYSE POUR LE GROUPE DES EXPERTS

En ne nous concentrant que sur le groupe des participants experts, on peut remarquer par l'intermédiaire de la figure 130 que la courbe d'apprentissage sur l'un ou l'autre des modes est beaucoup moins prononcée. Cette courbe d'apprentissage moins prononcée est logique dans le sens où un participant jugé expert est déjà censé être familiarisé avec les technologies de systèmes de réalité virtuelle ainsi que leurs problématiques associées. De plus un expert sera plus à l'aise à interagir avec un environnement virtuel et aura une meilleure notion de l'espace 3D.

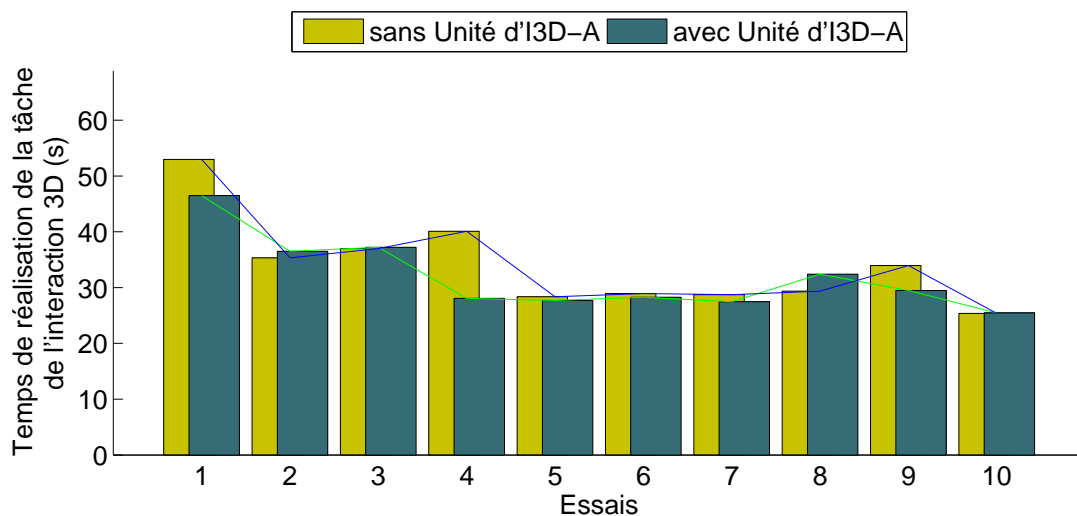


Figure 130: Temps moyen de la réalisation de l'épreuve - Groupe des experts.

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	31.91	41.59	6.45	25.48	46.48
MANUEL	34.01	65.41	8.09	25.38	52.97

Table 14: Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Groupe des experts.

Le tableau 14 nous amène au même constat que précédemment : la moyenne de temps est quasiment la même pour les deux modes et la variance est sensiblement plus faible pour le mode automatique que pour le mode manuel. Ceci démontre que même pour le groupe des experts, l'utilisation de l'unité d'interaction 3D adaptative permet de "lisser" le basculement d'une tâche de l'interaction 3D à l'autre.

C - ANALYSE POUR LE GROUPE DES INTERMÉDIAIRES

A la vue du graphique 131, le groupe des intermédiaires est sans aucun doute le groupe qui a le plus de difficulté à interagir avec les deux modes de basculements. Le temps moyen est quasiment identique entre les deux modes, mais on aperçoit que l'écart-type est beaucoup plus élevé pour le mode automatique que pour le mode manuel.

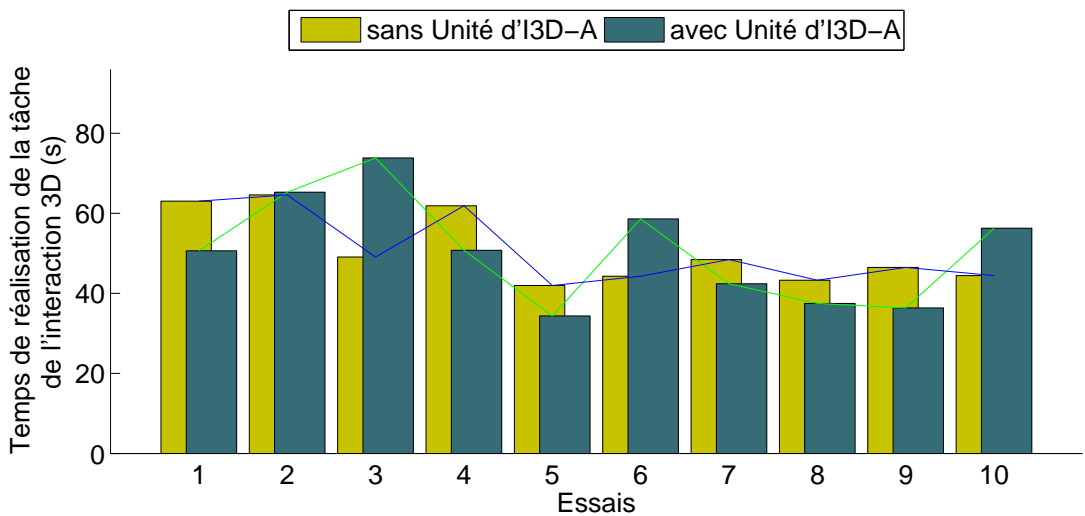


Figure 131: Temps moyen de la réalisation de l'épreuve - Groupe des intermédiaires.

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	50.61	173.01	13.15	34.36	73.83
MANUEL	50.76	78.72	8.87	41.97	64.62

Table 15: Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Groupe des intermédiaires.

D - ANALYSE POUR LE GROUPE DES NOVICES

La figure 132 et le tableau 16 démontrent que pour le groupe des novices, l'unité d'interaction 3D adaptative n'influe pas sur le temps moyen de réalisation de l'épreuve (50.52 avec, 50.82 sans), cependant l'écart-type est plus faible avec l'unité témoignant que les utilisateurs réalisaient l'épreuve avec des temps plus réguliers que sans l'unité.

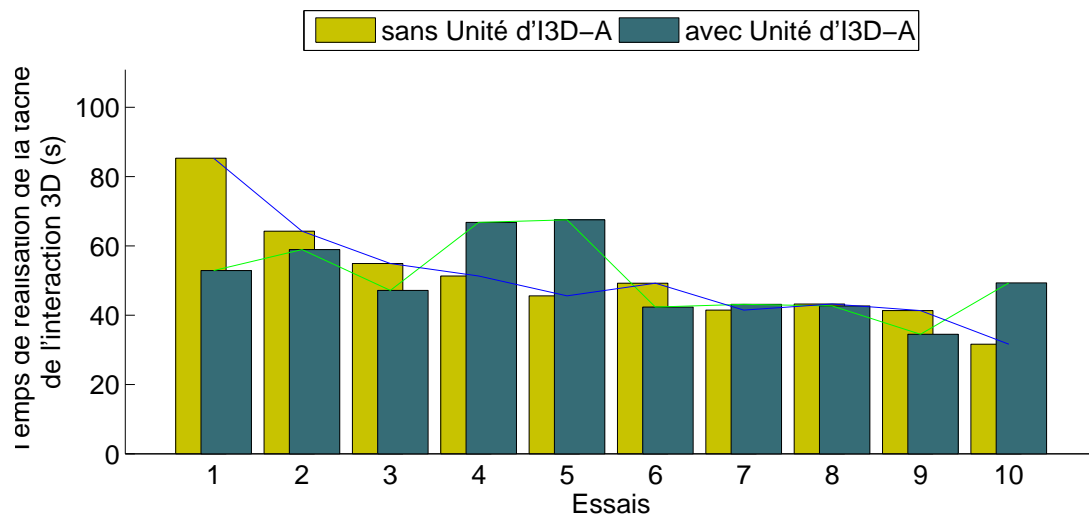


Figure 132: Temps moyen de la réalisation de l'épreuve - Groupe des novices.

MODE	MOYENNE	VARIANCE	ECART-TYPE	MIN	MAX
AUTOMATIQUE	50.52	120.29	10.97	34.48	67.52
MANUEL	50.82	224.39	14.98	31.64	85.30

Table 16: Valeurs caractéristiques des temps moyens de la réalisation de l'épreuve - Groupe des novices.

6.3.4 Soumission du questionnaire d'évaluation

Après chaque évaluation, l'utilisateur se voit remettre un questionnaire d'évaluation subjective. Ce questionnaire peut être lu en annexe A. Il a été conçu à partir de ceux proposés par Hrimech [Hrimech, 2009]. Ce questionnaire contient à la fois des informations sur le profil de l'utilisateur, que sur le jugement de celui-ci sur le mode basculement automatique et manuel.

6.3.5 Etude de l'utilisabilité

A la question 13 du questionnaire, les participants ont majoritairement répondu que la détection de la tâche par l'unité d'interaction 3D adaptative était rarement erronée. Certains même ont répondu que le système ne s'était jamais trompé, alors que l'observation de leurs résultats laissait apparaître que l'unité d'interaction 3D adaptative s'était trompée au moins une fois.

A la vue de l'histogramme, représenté figure 136, on peut conclure que la totalité des participants, à l'exception d'un seul, ont jugé l'accomplissement de l'épreuve facile, voire très facile avec l'unité d'interaction 3D adaptative. En revanche, les trois quarts des participants ont jugé difficile voir normal l'accomplissement de l'épreuve sans

l'unité d'interaction 3D adaptative. Cela démontre que l'unité d'interaction 3D adaptative a été grandement appréciée par les utilisateurs et qu'ils la considèrent comme une aide.

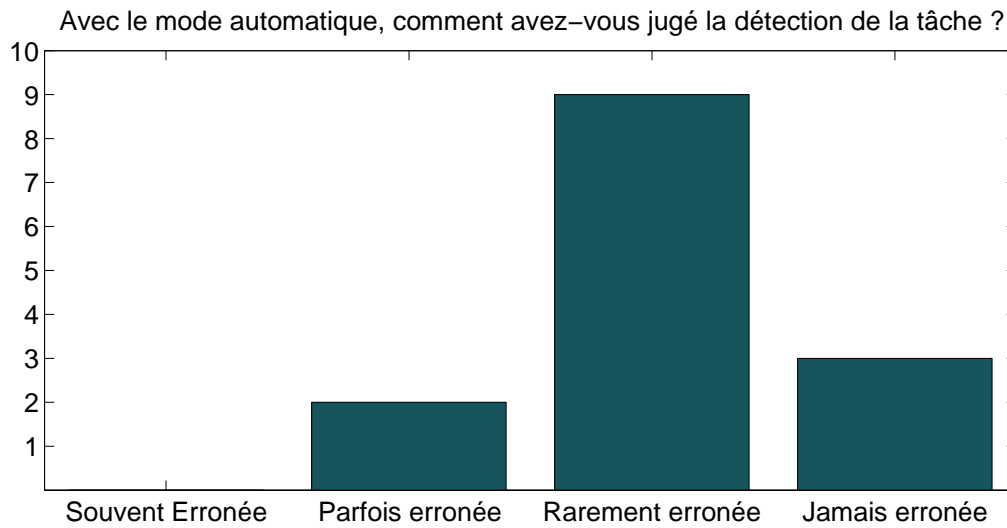


Figure 133: Jugement de la fiabilité de l'unité d'interaction 3D adaptative par les participants.

Ces résultats sont confirmés par le graphique 134 représentant les résultats des questions 12 et 14, sur l'aspect pratique du basculement de la tâche de l'interaction 3D avec ou sans unité d'interaction 3D adaptative. On retrouve à peu près la même distribution de réponse que pour la question précédente, démontrant ainsi que la quasi-totalité des participants considèrent comme pratique voire très pratique l'unité d'interaction 3D adaptative. Il faut néanmoins noter, que l'ensemble des participants juge le mode manuel comme assez pratique, ce qui signifie que le mode manuel leur semblait assez aisé à utiliser et qu'ils étaient satisfaits de son fonctionnement.

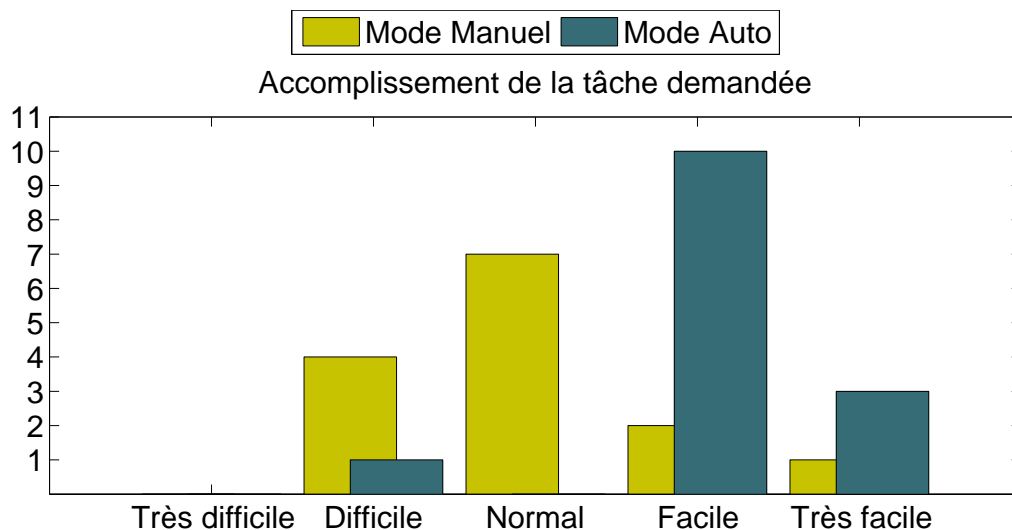


Figure 134: Jugement des participants sur la difficulté à accomplir l'épreuve de l'évaluation.

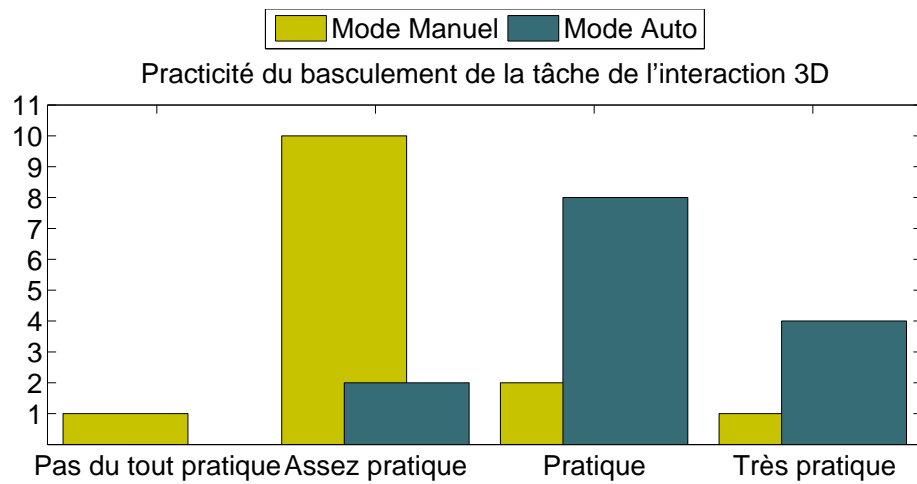


Figure 135: Jugement des participants sur la praticité à basculer d'une tâche de l'interaction 3D à une autre.

6.3.6 Etude du confort

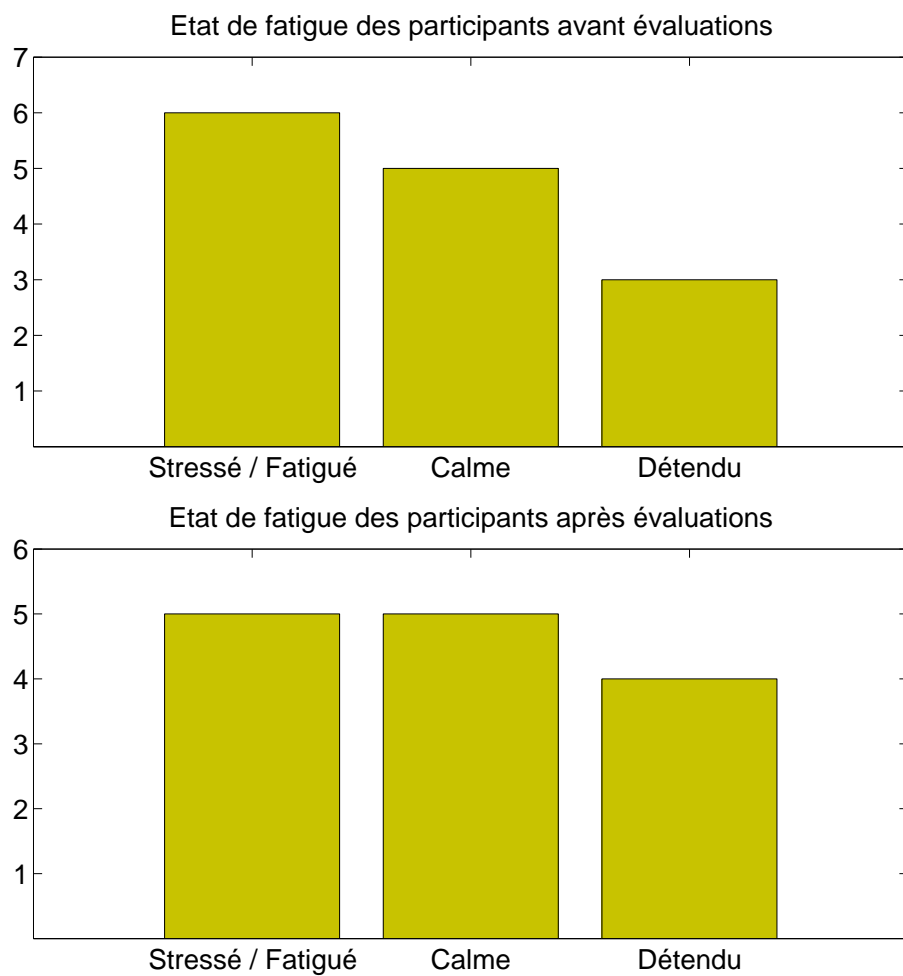


Figure 136: Etat de fatigue des participants avant et après l'évaluation (recueillis à partir des questionnaires.)

Les histogrammes représentés en figure 136 représentent l'état de fatigue des participants avant et après évaluation. Leur état de fatigue n'a pas évolué sauf pour une personne qui est passée de calme à détendue. La technique d'interaction 3D Fly-Over ne leur a pas causé de fatigue cognitive, démontrant ainsi une bonne réaction des participants vis-à-vis de cette technique.

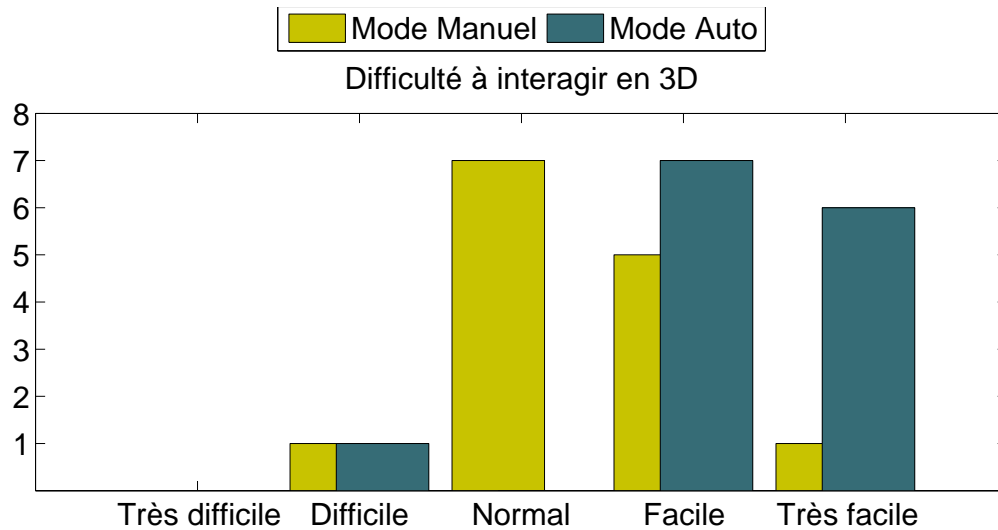


Figure 137: Jugement des participants sur l'interaction 3D.

La figure 137 permet de montrer la difficulté qu'ont éprouvée les participants à interagir en 3D avec Fly-Over avec ou sans l'unité d'interaction 3D adaptative. Là encore les résultats sont homogènes avec les précédents, c'est-à-dire que la majorité éprouve un jugement plus positif en mode automatique qu'en mode manuel.

6.4 OBSERVATIONS CONSTATÉES PENDANT LES ÉVALUATIONS

1. La quasi-totalité des participants ont, après seulement quelques essais, apprécié le mode automatique dans la mesure où même lorsque l'unité d'interaction 3D adaptative était désactivée, on pouvait les surprendre à attendre le basculement automatique de la tâche de l'interaction 3D.
2. Une grande partie des participants ont relâché leur attention durant les évaluations.
3. Certains utilisateurs ne se sont même pas aperçus que l'unité d'interaction 3D adaptative s'était trompée de tâche lors de certains basculements.
4. Si l'on regarde attentivement tous les graphiques représentant l'évolution de certains paramètres au fur et à mesure des essais, on remarquera qu'en moyenne le 4^{ème} ou 5^{ème}, ainsi que le 10^{ème} essai, la valeur de ces paramètres étaient beaucoup plus élevées. Nous avons expliqué ce phénomène, par un manque de concentration et un relâchement de l'attention de la part participants vers la moitié et la fin des évaluations. Ce phénomène a été confirmé par certains participants auxquels on a posé la question.

6.5 CONCLUSION

Dans ce chapitre nous nous sommes intéressés à l'évaluation de notre unité d'interaction 3D adaptative, qui rappelons-le, permet un basculement automatique d'une tâche de l'interaction 3D vers une autre en fonction du comportement de l'utilisateur.

Nous avons dans un premier temps, proposé un protocole d'évaluation rigoureux permettant d'isoler au maximum les paramètres à observer de manière à pouvoir effectuer une bonne analyse de l'évaluation. Ensuite, nous avons fait passer une quinzaine de participants sur ce protocole d'évaluation. L'analyse des paramètres observés ont pu démontrer que l'unité d'interaction 3D adaptative s'est avérée très efficace dans le remplacement de la décision humaine pour le problème du basculement de la tâche de l'interaction 3D, qui rappelons le est loin d'être quelque chose de naturel dans le domaine de la réalité virtuelle.

De plus, à la vue des questionnaires distribués à l'ensemble des participants lors de la phase d'évaluation subjective, une majorité des utilisateurs a jugé ce mode de basculement comme fiable et qu'elle n'a pas eu de difficulté à interagir en 3D avec ce mode de basculement. Nous pouvons donc conclure quant à l'efficacité de notre unité d'interaction 3D adaptative, pour le basculement d'une tâche de l'interaction 3D à une autre.

Partie III

CONCLUSION & PERSPECTIVES

CONCLUSION GÉNÉRALE

Mes travaux de recherche ont porté sur la proposition, la réalisation et l'évaluation d'un système d'interaction 3D adaptative, c'est-à-dire un système permettant à un utilisateur d'interagir de manière continue avec un environnement virtuel. Cette recherche s'est axée autour de deux domaines de compétences : le traitement de données et la réalité virtuelle. Cette approche nous a permis de bénéficier de chacun des avantages de ces deux domaines complémentaires, qui ne s'associent que trop rarement.

Le sujet est composé de trois parties couvrant la totalité d'un système de réalité virtuelle, en partant des périphériques même, jusqu'à l'interaction 3D même de l'utilisateur avec l'environnement virtuel. La qualité et la continuité de l'interaction 3D étant dépendante du système de tracking utilisé, il était essentiel de s'intéresser de près au traitement des données issues du système tracking.

L'état de l'art a été orienté vers trois sujets. Nous nous sommes donc intéressés dans un premier temps, au domaine de la réalité virtuelle, incluant les métaphores et techniques couramment utilisées, ainsi que les architectures que l'on retrouve communément dans ce domaine. Puis, nous avons étudié les différents périphériques utilisés en réalité virtuelle pour faire du suivi. Nous avons souligné, leurs points forts et points faibles ainsi que les contraintes qu'ils imposaient. Enfin, nous avons présenté les protocoles et méthodes de calibration de ces systèmes. Ainsi que les outils dont nous disposons afin de caractériser ces périphériques et par la suite, prendre la meilleure décision pour effectuer la calibration.

A l'issue de cet état l'art nous avons pu constater certains points. Dans le domaine de l'interaction 3D, nous avons pu voir qu'il n'existait pas de technique générique pour réaliser les 3 tâches de l'interaction 3D. Nous avons observé que bien souvent les techniques d'interaction 3D sont dépendantes du matériel, voire conçue pour celui-ci. Le principal problème étant que les différentes tâches de l'interaction 3D sont temporellement séparées et que le passage de l'une à l'autre se fait bien souvent par décision humaine par le biais du contrôle d'application (pression sur un bouton, reconnaissance vocale ou gestuelle) et que, par conséquent, une interaction 3D continue est impossible. Dans le domaine des architectures de systèmes de réalité virtuelle, force est de constater, qu'elles ne sont généralement qu'un amalgame de différents composants, plus ou moins conçus pour une application en réalité virtuelle. De plus, bien souvent, ces architectures prennent le parti de privilégier un aspect bien particulier de l'interaction 3D et non de traiter la globalité du système de réalité virtuelle. Enfin, pour ce qui concerne les approches pour la calibration de systèmes de tracking, nous avons vu qu'il était primordial de bien caractériser ceux-ci, par le biais d'un protocole rigoureux, avant de choisir une approche pour le processus de calibration. A la lecture des différents travaux de recherche, nous avons pris conscience que parmi les innombrables méthodes mathématiques existantes, il n'en existait pas une plus efficace que les autres, mais que l'on devait choisir l'approche au cas par cas, à la vue des résultats de la caractérisation.

La partie contribution de cette thèse, peut être séparée en trois sous parties : traitant chacune d'un aspect bien particulier d'un système de réalité virtuelle.

La première présente une *technique d'interaction 3D innovante : Fly-Over*, permettant d'accomplir toutes les tâches de l'interaction 3D et qui soit compatible avec la plupart des périphériques de tracking. Cette technique est totalement innovante dans le sens où on définit des zones d'interaction 3D autour de la main de l'utilisateur et qu'elle permet de passer d'une tâche de l'interaction 3D à l'autre, tout en conservant la même approche pour l'utilisateur. Sa composante navigation est une création entièrement personnelle. Les composantes de sélection et de manipulation étant quant à elles basées sur des techniques existantes, mais avec cependant quelques évolutions que nous avons proposées afin de les intégrer de manière transparente dans le continuum de l'interaction 3D.

La seconde partie, quant à elle, présente ce que nous avons appelée l'*unité d'interaction 3D adaptative*, qui est un composant permettant au système de déterminer quelle tâche de l'interaction 3D (navigation, sélection ou manipulation) l'utilisateur souhaite réaliser et donc de basculer d'une tâche de l'interaction 3D à l'autre automatiquement. Cette unité se base sur l'observation de quatre paramètres inhérents à la technique d'interaction 3D *Fly-Over*, sur lesquels est appliquée une table de décision permettant d'estimer la tâche d'interaction 3D souhaitée. Ce composant garantit la continuité de l'interaction 3D en se substituant au système de commandes qui est généralement géré par l'utilisateur et donc par décision humaine. Une évaluation de ce composant, présentée dans le chapitre 6 a démontré que le système pouvait se tromper sur la détermination de la tâche de l'interaction 3D que souhaitait faire l'utilisateur, cependant cela ne nuisait pas à l'accomplissement de l'épreuve demandée. En effet, lors de l'évaluation nous n'avons pas remarqué pas d'augmentation du temps de réalisation de l'épreuve avec l'unité d'interaction 3D adaptative. Cette évaluation nous a donc permis de justifier un véritable apport à l'interaction 3D en prouvant que ce composant pouvait se substituer à la décision humaine sans pour autant dégrader les performances de l'utilisateur.

La dernière partie, quant à elle, traite de la *calibration d'un système de tracking* peu précis : le SPIDAR, par rapport à un autre système plus précis, le tracking optique afin de garantir la continuité des données. Dans un premier temps, nous avons répertorié l'ensemble des problèmes de conceptions du SPIDAR. Puis, dans un second temps, nous avons proposé un protocole expérimental afin de caractériser le SPIDAR et mettre en exergue les problèmes inhérents au SPIDAR. Ce protocole est innovant, dans la mesure où nous avons utilisé la réalité virtuelle pour relever un ensemble de mesures dans un espace de travail défini à 1 m³ et de manière homogène. Suite à l'étude des mesures prélevées grâce à ce protocole, nous avons jugé l'utilisation d'un réseau de neurones comme solution la plus appropriée pour accomplir cette tâche. Néanmoins, nous avons quand même évalué la calibration du SPIDAR par le biais d'autres approches telles que l'interpolation linéaire et les SVRs. Suite à la calibration du SPIDAR par le réseau de neurones, nous avons grandement diminué les erreurs absolues sur sa position passant ainsi d'une erreur absolue moyenne de 72.86 mm à 7.80 mm. Afin de garantir, l'efficacité de la calibration réalisée, à chaque démarrage du SPIDAR, nous avons proposé un système d'initialisation multimodale pour le SPIDAR afin de permettre à un utilisateur d'initialiser correctement le SPIDAR. Par le biais de ce système, l'utilisateur peut aisément et précisément placer l'effecteur au centre la structure du SPIDAR. Il dispose du retour d'effort sur l'effecteur qui incite l'effecteur à converger vers le point d'initialisation,

ainsi que d'un graphique visuel et d'un retour sonore lui permettant de savoir s'il est proche ou non du point d'initialisation. La détermination du point d'initialisation se réalise avec une précision de l'ordre de 1.2 mm

Afin d'appliquer les multiples traitements de données dans l'ensemble du système de réalité virtuelle, comme par exemple : la calibration du SPIDAR et permettre un basculement d'un système de tracking à l'autre, nous avons conçu un composant, appelé : *unité de correction et de combinaison de données*, permettant, à partir d'un fichier de description, d'appliquer des traitements à la volée aux données issues des différents périphériques.

L'ensemble de ce manuscrit témoigne de la volonté de concevoir un système d'interaction 3D en ayant pris en compte tous les aspects inhérents à un système de réalité virtuelle. En résumé, le travail réalisé a présenté des originalités tout au long de cette thèse. La gestion des périphériques, la technique d'interaction 3D cohérente pour les différentes tâches, les nombreuses assistances à l'interaction 3D et l'apport de traitements de données dans tous les composants du système, nous ont donc permis de garantir une interaction 3D continue, tant au niveau logicielle que matérielle. Nous avons aussi pu décharger l'utilisateur de la gestion du système de commande (ou du contrôle d'application) par le biais de l'unité d'interaction 3D adaptative. Par ce dernier point, nous avons transformé le système de réalité virtuelle en système adaptatif, le rendant capable de s'adapter au contexte et de prendre certaines décisions à la place de l'utilisateur ou du moins d'estimer ses intentions. Ceci nous amène aux perspectives envisagées pour faire suite à ces travaux de recherche.

Cette thèse a été le fruit d'une collaboration étroite entre le domaine de la réalité virtuelle et du traitement de données. Nous pensons que cette collaboration doit perdurer car ces deux domaines ont beaucoup de choses à apprendre l'un de l'autre et cette thèse en est la preuve.

La technique d'interaction 3D Fly-Over a été évaluée seulement pour sa composante navigation, mais il est intéressant d'évaluer aussi ses composantes : sélection et manipulation, dans un scénario élaboré regroupant plusieurs successions des tâches de l'interaction 3D. Dans ce même scénario, on pourrait également améliorer l'évaluation de l'unité d'interaction 3D adaptative.

L'unité d'interaction 3D adaptative peut notamment être plus indépendante de la technique d'interaction 3D Fly-Over, car à l'heure actuelle, les paramètres observés sont grandement liés à cette technique. Il serait intéressant de pouvoir utiliser cette unité avec d'autres techniques.

La technique d'interaction 3D Fly-Over, gagnerait à se voir octroyer plus de reconnaissance de gestes ou bien des zones d'interaction 3D supplémentaires afin de lui permettre de répondre à plus de problématiques. Ainsi on pourrait imaginer des zones d'interaction 3D pour la tâche de manipulation, ou bien utiliser des gestes de manipulation multipoints (multitouch), afin d'effectuer des mises à l'échelle ou bien des rotations précises.

De nombreuses améliorations peuvent être également apportées au SPIDAR, que ce soit d'un point de vue mécanique (fixer les encodeurs sur une poulie indépendante de l'axe moteur) ou du point de vue de sa calibration. En effet, en utilisant certains principes issus du domaine de l'automatique on pourrait asservir de manière plus efficace la commande de l'effecteur du SPIDAR afin de contrôler celui-ci avec plus de sensibilité et ainsi transformer la phase d'initialisation semi-automatique en automatique. De plus, en utilisant une approche hybride pour la calibration, c'est-à-dire l'utilisation de plusieurs méthodes mathématiques, (éventuellement réseau de neurones avec SVRs) il est sans doute possible d'augmenter un peu plus la précision du SPIDAR et ainsi, pourquoi ne pas se passer totalement du tracking optique et s'appuyer seulement sur le tracking fourni par le SPIDAR.

Partie IV

BIBLIOGRAPHIE

BIBLIOGRAPHIE

- Abásolo M, Della J (2007) Magallanes: 3d navigation for everybody. Proceedings of the 5th international conference on Computer graphics and interactive techniques URL <http://portal.acm.org/citation.cfm?id=1321261.1321285> (Cité aux pages [xiv](#) et [17](#).)
- Adachi S, Ogawa T (2002) A new system identification method based on support vector machines. Proceedings of the international conference on Adaptation and Learning in Control and Signal Processing: IFAC Workshop p 181, URL http://scholar.google.com.gate6.inist.fr/scholar?hl=en&lr=&q=info:yMJGhLAY5_kJ:scholar.google.com/&output=search (Cité aux pages [62](#) et [64](#).)
- Advance Real-Time Tracking GmbH (2010) <http://www.ar-tracking.de/> (Cité à la page [42](#).)
- An J, Yang Q, Ma Z (2007) Study on svm on-line function regression method for mass data. Machine Learning and Cybernetics 5:2773–2777, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4370619 (Cité à la page [64](#).)
- Andújar C, Vázquez P, Fairén M (2004) Wayâfinder: guided tours through complex walkthrough models. Computer Graphics Forum 23(3):499–508, URL <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2004.00781.x/full> (Cité à la page [29](#).)
- Barratt D, Davies A, Hughes A, Thom S (2001) Accuracy of an electromagnetic three-dimensional ultrasound system for carotid artery imaging. Ultrasound in medicine & biology 27(10):1421–1425, URL <http://linkinghub.elsevier.com/retrieve/pii/S0301562901004471> (Cité aux pages [xv](#), [xvi](#), [55](#), et [68](#).)
- Beckhaus S, Ritter F, Strothotte T (2002) Cubicalpath-dynamic potential fields for guided exploration in virtual environments. Proceedings The Eighth Pacific Conference on Computer Graphics and Applications pp 387–459, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=883963 (Cité à la page [29](#).)
- Birkfellner W, Watzinger F, Wanschitz F, Enislidis G, Truppe M, Ewers R, Bergmann H (1998) Concepts and results in the development of a hybrid tracking system for cas. Lecture Notes in Computer Science: Medical Image Computing and Computer-Assisted Intervention 1496/1998:343, URL <http://scholar.google.com.gate6.inist.fr/scholar?hl=en&lr=&q=info:wUXgSgoqqH0J:scholar.google.com/&output=search> (Cité aux pages [xvi](#), [70](#), et [132](#).)
- Bowman D (1999) Interaction techniques for common tasks in immersive virtual environments. Thèse de doctorat de l'Institut de Technologie de Georgie URL <http://visac.acmc.uq.edu.au/pdfs/InteractionTechinVE.pdf> (Cité aux pages [xiv](#), [14](#), [17](#), [21](#), [22](#), et [24](#).)

- Bowman D (2004) 3d user interfaces: Theory and practice. Addison-Wesley Educational Publishers Inc URL <http://books.google.com/books?id=It5QAAAAMAAJ&printsec=frontcover> (Cité aux pages 11, 20, et 22.)
- Bowman D, Hodges LF (1997) An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. Proceedings of the symposium on Interactive 3D graphics pp 35–38, URL <http://portal.acm.org/citation.cfm?id=253301> (Cité à la page 22.)
- Bryson S (1992) Measurement and calibration of static distortion of position data from 3d trackers. Proceedings of the Special Interest Group on Graphics and Interactive Techniques URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.8250&rep=rep1&type=pdf> (Cité aux pages xvi, 66, et 68.)
- Burdea G, Coiffet P (1993) La réalité virtuelle. Hermès URL <http://books.google.com/books?id=sPj6PAAACAAJ&printsec=frontcover> (Cité aux pages xiv et 9.)
- Chávez F, Vega F, Olague G, Montero J (2008) An independent and non-intrusive laser pointer environment control device system. Proceedings of the international conference on pervasive services pp 37–46, URL <http://portal.acm.org/citation.cfm?id=1387269.1387276> (Cité aux pages xv, 43, et 44.)
- Chung J (1993) Intuitive navigation in the targeting of radiation therapy treatment beams. Thèse de doctorat de l'Université de Caroline du Nord, USA URL <http://portal.acm.org/citation.cfm?id=194062> (Cité à la page 16.)
- Cyber Gloves Systems (2010a) Cybergloves. <http://www.cyberglovesystems.com/products/cyberglove-ii/overview> (Cité aux pages xv et 46.)
- Cyber Gloves Systems (2010b) Cybergrasp. <http://www.cyberglovesystems.com/products/cybergrasp/overview> (Cité aux pages xv, 46, et 47.)
- Cyber Gloves Systems (2010c) Cybertouch. <http://www.cyberglovesystems.com/products/cybertouch/overview> (Cité aux pages xv et 47.)
- Dassault Systèmes (2010) Virtools. <http://www.virtools.com/> (Cité à la page 32.)
- Fifth Dimension Technologies (2010) 5dt data glove ultra serie. <http://www.5dt.com/> (Cité aux pages xv et 46.)
- Fitzgibbon A (2003) Robust registration of 2d and 3d point sets. Image and Vision Computing 21(13-14):1145–1153, URL <http://linkinghub.elsevier.com/retrieve/pii/S0262885603001835> (Cité à la page 69.)
- Fitzmaurice G, Matejka J, Mordatch I, Khan A (2008) Safe 3d navigation. Proceedings on the 2008 Symposium on Interactive 3D graphics and games pp 7–16, URL <http://portal.acm.org/citation.cfm?id=1342252> (Cité à la page 29.)
- Fuchs P, Moreau G (2003) Le traité de la réalité virtuelle: Fondements et interfaces comportementales. Les Presses de l'Ecole des mines 1, URL <http://books.google.com/books?id=CrM9NAAACAAJ&printsec=frontcover> (Cité aux pages xiv, 9, 10, 11, et 12.)

- Fuhrmann T, Klein M, Odendahl M (2003) Bluewand-a versatile remote control and pointing device. Proceedings of the Kommunikation in Verteilten Systemen, Kurzbeiträge: Workshop pp 81–88, URL <http://www.google.com/search?client=safari&rls=en-us&q=Bluewand-A+versatile+remote+control+and+pointing+device&ie=UTF-8&oe=UTF-8> (Cité aux pages xv et 50.)
- Ghazisaedy M, Adamczyk D (1995) Ultrasonic calibration of a magnetic tracker in a virtual reality space. Proceedings of the Virtual Reality Annual International Symposium URL <http://doi.ieeecomputersociety.org/10.1109/VRAIS.1995.512494> (Cité aux pages xv, xvi, 65, 66, et 67.)
- Hachet M, Decle F, Knödel S, Guitton P (2009) Navidget for 3d interaction: Camera positioning and further uses. International Journal of Human-Computer Studies 67(3):225–236 (Cité aux pages xiv, 18, et 19.)
- Haption (2010) Virtuouse. <http://www.haption.com/site/fr/html/materiel.php?item=1> (Cité aux pages xv et 48.)
- Hernandez-Rebollar J, Kyriakopoulos N (2002) The acceleglove: a whole-hand input device for virtual reality. Proceeding of the International Conference on Computer Graphics and Interactive Techniques pp 259 – 259, URL <http://portal.acm.org/citation.cfm?id=1242272> (Cité aux pages xv et 47.)
- Hrimech H (2009) Evaluation de métaphores d’interaction pour le travail collaboratif entre sites distants d’immersion virtuelle. Thèse de doctorat de l’École Nationale Supérieure d’Arts et Métiers URL <http://paristech.bib.rilk.com/5576/> (Cité à la page 146.)
- Ikits M, Hansen C, Johnson C (2003) A comprehensive calibration and registration procedure for the visual haptic workbench. Proceedings of the workshop on Virtual environments 39:247–254, URL <http://portal.acm.org/citation.cfm?id=769982> (Cité aux pages xvi, 67, et 68.)
- Irawati S, Calderón D, Ko H (2005) Semantic 3d object manipulation using object ontology in multimodal interaction framework. Proceedings of the International Conference on Artificial Reality and Tele-existence p 39 (Cité aux pages xiv et 27.)
- Irawati S, Calder D, Ko H (2006) Spatial ontology for semantic integration in 3d multimodal interaction framework. VRCIA pp 129–135 (Cité aux pages xv, 34, et 37.)
- Jacoby R, Adelstein B, Ellis S (1996) Improved temporal response in virtual environments through system hardware and software reorganization. Proceeding of the international society for optical engineering pp 271–284 (Cité à la page 20.)
- Just C, Bierbaum A, Baker A, Cruz-Neira C (1998) Vr juggler: A framework for virtual reality development. Proceedings of the Immersive Projection Technology Workshop URL <http://www.vision.auc.dk/education/teaching/e03/cvg9/vr/vr.juggler.framework.for.vr.dev.1998.IPT98.pdf> (Cité à la page 40.)
- Kaiser E, Olwal A, McGee D, Benko H, Corradini A, Li X, Cohen P, Feiner S (2003) Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. Proceedings of the international conference on Multimodal interfaces pp 12–19, URL <http://portal.acm.org/citation.cfm?id=958432.958438> (Cité aux pages xv, 38, et 39.)

- Kemp C, Anderson C, Nguyen H, Trevor A, Xu Z (2008) A point-and-click interface for the real world: laser designation of objects for mobile manipulation. Proceedings of the international conference on Human robot interaction pp 241–248, URL <http://portal.acm.org/citation.cfm?id=1349822.1349854> (Cité à la page 43.)
- Khan A, Mordatch I, Fitzmaurice G, Matejka J (2008) Viewcube: a 3d orientation indicator and controller. Proceedings of the 2008 symposium on Interactive 3D graphics and games pp 17–25, URL <http://portal.acm.org/citation.cfm?id=1342250.1342253> (Cité aux pages xiv et 25.)
- Kim S, Ishii M, Koike Y, Sato M (2000) Development of tension based haptic interface and possibility of its application to virtual reality. Proceedings of the symposium on Virtual reality software and technology pp 199–205 (Cité à la page 49.)
- Kim S, Koike Y, Sato M (2002) Tension based 7 dofs force feedback device: Spidar-g. Transactions on Control, Automation, and Systems Engineering 4(1):9–16 (Cité à la page 49.)
- Kindratenko V, Sherman W (2005) Neural network-based calibration of electromagnetic tracking systems. Virtual Reality 9(1):70–78 (Cité aux pages xvi et 69.)
- Kirstein C, Muller H (1998) Interaction with a projection screen using a camera-tracked laser pointer. Proceedings of the Conference on MultiMedia Modeling URL <http://doi.ieeecomputersociety.org/10.1109/MULMM.1998.723001> (Cité à la page 43.)
- König W, Böttger J, Völzow N, Reiterer H (2008) Laserpointer-interaction between art and science. Proceedings of the international conference on Intelligent user interfaces pp 423–424, URL <http://portal.acm.org/citation.cfm?id=1378773.1378848> (Cité à la page 43.)
- Kurz D, Hantsch F, Grobe M, Schiewe A, Bimber O (2007) Laser pointer tracking in projector-augmented architectural environments. ISMAR URL <http://e-pub.uni-weimar.de/volltexte/2007/874/pdf/LPTRE.pdf> (Cité à la page 43.)
- Laboratoire IBISC (2010) Evr@. <http://evra.ibisc.univ-evry.fr/index.php/La-Plate-Forme-technologique-Evr@> (Cité à la page 112.)
- Latoschik M, Bomberg E, Herder J (2006) Augmenting a laser pointer with a diffraction grating for monoscopic 6dof detection. Journal of Virtual Reality and Broadcasting 4(14), URL <http://www.jvrb.org/4.2007/1275/4200714.pdf> (Cité à la page 43.)
- Leonhardt U, Magee J (1998) Multi-sensor location tracking. Proceedings of the international conference on Mobile computing and networking pp 203–214 (Cité aux pages xv, 32, et 33.)
- Leotta D (2004) An efficient calibration method for freehand 3-d ultrasound imaging systems. Ultrasound in medicine & biology 30(7):999–1008, URL <http://linkinghub.elsevier.com/retrieve/pii/S0301562904001334> (Cité aux pages xv et 54.)
- Livingston M (1997) Magnetic tracker calibration for improved augmented reality registration. Presence: Teleoperators and Virtual Environment 6(5):532–546, URL <http://eprints.kfupm.edu.sa/49599/1/49599.pdf> (Cité aux pages xvi et 69.)

- Mackinlay J, Card S, Robertson G (1990) Rapid controlled movement through a virtual 3d workspace. Proceedings of the 17th annual conference on Computer graphics and interactive techniques pp 171–176, URL <http://portal.acm.org/citation.cfm?id=97879.97898> (Cité à la page 18.)
- Maidi M (2007) Suivi hybride en présence d’occultations pour la réalité augmentée. Thesis, University of Evry (Cité à la page 71.)
- Mapes D, Moshell J (1995) A two-handed interface for object manipulation in virtual environments. Presence: Teleoperators and Virtual Environments 4(4):403–416, URL <http://www.csa.com/partners/viewrecord.php?requester=gs&collection=TRD&recid=A9619649AH> (Cité à la page 18.)
- MathWorks (2010) Matlab®. <http://www.mathworks.fr/products/matlab/> (Cité aux pages 65, 102, et 103.)
- Matveyev S, Matveyev K (2004) Tracking of a laser pointer space orientation by using single camera. Proceedings of the International Conference on Artificial Reality and Tele-existence URL http://vrsj.ime.cmc.osaka-u.ac.jp/ic-at/papers/2004/Poster_Session/P-009.pdf (Cité aux pages xv et 44.)
- McCulloch W, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5:115–133, URL <http://www.google.com/search?client=safari&rls=en-us&q=A+logical+calculus+of+the+ideas+immanent+in+nervous+activity.&ie=UTF-8&oe=UTF-8> (Cité à la page 56.)
- Milgram P, Kishino F (1994) A taxonomy of mixed reality visual displays (special issue on networked reality). Institute of Electronics, Information and Communication Engineers Transactions on Information and Systems pp 1321–1329, URL <http://ci.nii.ac.jp/naid/110003209335/> (Cité aux pages xiv et 8.)
- Mine M (1995) Virtual environment interaction techniques. UNC Chapel Hill Computer Science Technical Report TR95-018 URL <http://www.cvmt.dk/education/teaching/e04/cvg9/VR/mine95virtual.pdf> (Cité aux pages xiv, 14, 15, 16, 18, et 19.)
- Nintendo® (2010) Wiimote controller. <http://www.nintendo.com/wii/what/controllers#remote> (Cité aux pages xv, 49, et 50.)
- Omnès R (1995) A new interpretation of quantum mechanics and its consequences in epistemology. Foundations of Physics 25(4):605–629, URL <http://www.springerlink.com/index/R21831R625248VR1.pdf> (Cité à la page 9.)
- Otmane S, Davesne F (2009) Utilization of human scale spidar-g in the framework of assistance to 3dinteraction in a semi-immersive ar/vr platform. Proceedings of the Joint Virtual Reality Conference EGVE-ICAT-EURO VR (JCVR 2009) URL <http://hal.archives-ouvertes.fr/hal-00423405/> (Cité aux pages xv, 28, 29, et 30.)
- Otmane S, Ouramdane N, Mallem M (2007) Towards a collaborative 3d interaction model for cooperative design in virtual environments. Proceeding of the International Conference on Computer Supported Cooperative Work in Design pp 198–203, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4281434 (Cité aux pages xiv, 12, 27, et 28.)

- Ouramdane N, Davesne F, Otmane S, Mallem M (2006a) 3d interaction technique to enhance telemanipulation tasks using virtual environments. *International Conference on Intelligent Robots and Systems* pp 5201–5207 (Cité à la page 23.)
- Ouramdane N, Otmane S, Davesne F, Mallem M (2006b) Follow-me: a new 3d interaction technique based on virtual guides and granularity of interaction. *VRCIA* pp 137–144 (Cité à la page 23.)
- Ouramdane N, Otmane S, Mallem M (2009) Interaction 3d en réalité virtuelle-etat de l'art. *Technique et Sciences Informatiques* 28(8):1017–1049, URL <http://hal.archives-ouvertes.fr/hal-00376458/> (Cité aux pages 11 et 15.)
- Pausch R, Burnette T, Brockway D, Weiblen M (1995) Navigation and locomotion in virtual worlds via flight into hand-held miniatures. *Proceedings of the annual conference on Computer graphics and interactive techniques* pp 399–400, URL <http://portal.acm.org/citation.cfm?id=218495&dl=GUIDE&coll=GUIDE&CFTOKEN=71323902> (Cité aux pages xiv, 22, et 23.)
- Pierce J, Stearns B, Pausch R (1999) Voodoo dolls: seamless interaction at multiple scales in virtual environments. *Proceedings of the symposium on Interactive 3D graphics* pp 141–145, URL <http://portal.acm.org/citation.cfm?id=300523.300540> (Cité à la page 20.)
- Poupyrev I, Billinghurst M, Weghorst S, Ichikawa T (1996) The go-go interaction technique: non-linear mapping for direct manipulation in vr. *Proceedings of the annual symposium on User interface software and technology* pp 79–80, URL <http://portal.acm.org/citation.cfm?id=237091.237102> (Cité aux pages xiv et 21.)
- Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* 65(6), URL <http://www.ling.upenn.edu/courses/cogs501/Rosenblatt1958.pdf> (Cité à la page 56.)
- Sensable Technologies (2010) Phantom desktop. <http://www.sensable.com/haptic-phantom-desktop.htm> (Cité aux pages xv et 48.)
- Steinicke F, Ropinski T, Hinrichs K (2005) A generic virtual reality software system's architecture and application. *Proceedings of the International Conference on Artificial Reality and Tele-existence* pp 220–227, URL <http://portal.acm.org/citation.cfm?id=1152399.1152440> (Cité aux pages xv et 32.)
- Stoakley R, Conway MJ, Pausch R (1995) Virtual reality on a wim: interactive worlds in miniature. *Proceedings of the SIGCHI conference on Human factors in computing systems and graphics interface* URL <http://portal.acm.org/citation.cfm?id=223938> (Cité à la page 22.)
- Tan D, Robertson G, Czerwinski M (2001) Exploring 3d navigation: combining speed-coupled flying with orbiting. *Proceedings of the SIGCHI conference on Human factors in computing systems* pp 418–425, URL <http://portal.acm.org/citation.cfm?id=365307> (Cité à la page 18.)
- Taylor R, II, Hudson T, Seeger A, Weber H, Juliano J, Helser A (2001) Vrpn: a device-independent, network-transparent vr peripheral system. *Proceedings of the symposium on Virtual reality software and technology* pp 55 – 61, URL <http://portal.acm.org/citation.cfm?id=505008.505019> (Cité à la page 40.)

- Tisseau J (2001) *Réalité virtuelle: autonomie in virtuo*. Habilitation à diriger des recherches de l'université de Rennes, France URL <http://www.enib.fr/~tisseau/doc/hdr/hdrJT.pdf> (Cité aux pages [xiv](#), [9](#), et [10](#).)
- Tisseau J (2004) *Virtual reality and complexity*. Manifeste scientifique de l'Ecole Nationale d'Ingénieurs de Brest, France URL <http://www.enib.fr/~tisseau/pdf/paper/manifeste.uk.pdf> (Cité aux pages [xiv](#) et [10](#).)
- Ullah S, Ouramdane N, Otmane S, Richard P, Davesne F, Mallem M (2008) Augmenting 3d interactions with haptic guide in a large scale virtual environment. *Proceedings of The International Conference on Virtual-Reality Continuum and Its Applications in Industry* URL <http://portal.acm.org/citation.cfm?id=1477862.1477891> (Cité aux pages [xv](#) et [30](#).)
- Vapnik V (2000) *The nature of statistical learning theory*. Springer-Verlag New York, Inc (Cité aux pages [62](#) et [63](#).)
- Vogt F, Wong J, Fels S, Cavens D (2003) Tracking multiple laser pointers for large screen interaction. *Extended Abstracts of the ACM Symposium on User Interface Software and Technology* pp 95–96, URL <http://hct.ece.ubc.ca/publications/pdf/vogt-et-al-uist2003.pdf> (Cité à la page [43](#).)
- Widrow B, Angell J (1962) Reliable, trainable networks for computing and control. *Aerospace Engineering* 21(9):78–123, URL <http://www-isl.stanford.edu/~widrow/papers/j1962reliabletrainable.pdf> (Cité à la page [56](#).)
- Wilkes C, Bowman D (2008) Advantages of velocity-based scaling for distant 3d manipulation. *Proceedings of the symposium on Virtual reality software and technology* pp 23–29, URL <http://portal.acm.org/citation.cfm?id=1450579.1450585> (Cité à la page [22](#).)
- Zendjebil IM, Ababsa F, Didier JY, Lalagüe E, Decle F, Delmont R, Frauciel L, Vairon J (2009) *Réalité augmentée en extérieur: Etat de l'art*. *Technique et Science Informatiques (TSI), Réalité Virtuelle - Réalité Augmentée* 28(6-7/2009):857–890, numéro spécial (Cité à la page [71](#).)
- Zhang M, Yan W, Yuan Z (2004) Study of nonlinear system identification based on support vector machine. *Proceedings of the International Conference on Machine Learning and Cybernetics* 5:3287–3290, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1378604 (Cité à la page [64](#).)
- Zhao C, Han P (2007) Model identification of thermal object based on smooth support vector regression. *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition* 3(1388–1391), URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4421651 (Cité à la page [64](#).)

Partie V

ANNEXES

QUESTIONNAIRE

Evaluation Subjective

E - INFORMATIONS PERSONNELLES

1. Quel est votre nom ?
2. Quel est votre prénom ?
3. Quel âge avez-vous ?
4. Vous êtes : un homme - une femme
5. Vous êtes : Droitier - Gaucher - Ambidextre
6. Comment qualifieriez-vous votre niveau d'expérience en réalité virtuelle ? Novice
- Intermédiaire - Expert
7. Jouez-vous aux jeux-vidéo ?
 - a) ☐Jamais
 - b) ☐Parfois
 - c) ☐Souvent
 - d) ☐Très régulièrement
8. Comment jugiez-vous votre état avant l'évaluation ?
 - a) ☐Stressé / Fatigué
 - b) ☐Calme
 - c) ☐Détendu
9. Comment jugiez-vous votre état après l'évaluation ?
 - a) ☐Stressé / Fatigué
 - b) ☐Calme
 - c) ☐Détendu

F - UTILISABILITÉ

10. Avec le mode de basculement automatique, vous avez trouvé l'accomplissement de la tâche demandée :
 - a) ☐ Très difficile
 - b) ☐ Difficile
 - c) ☐ Normal
 - d) ☐ Facile
 - e) ☐ Très facile
11. Avec le mode de basculement manuel, vous avez trouvé l'accomplissement de la tâche demandée :
 - a) ☐ Très difficile
 - b) ☐ Difficile
 - c) ☐ Normal
 - d) ☐ Facile
 - e) ☐ Très facile
12. Dans le mode automatique de basculement de la tâche de l'interaction 3D, vous jugez ce mode de fonctionnement comme :
 - a) ☐ Pas du tout pratique
 - b) ☐ Assez pratique
 - c) ☐ Pratique
 - d) ☐ Très pratique
13. Dans le mode automatique de basculement de la tâche de l'interaction 3D, vous jugez la détection de la tâche que vous souhaitiez accomplir comme :
 - a) ☐ Souvent erronée
 - b) ☐ Parfois erronée
 - c) ☐ Rarement erronée
 - d) ☐ Jamais erronée

-
14. Dans le mode manuel de basculement de la tâche de l'interaction 3D, vous jugez ce mode de fonctionnement comme :
- a) ☐ Pas du tout pratique
 - b) ☐ Assez pratique
 - c) ☐ Pratique
 - d) ☐ Très pratique
15. Dans le mode manuel de basculement de la tâche de l'interaction 3D, vous jugez la facilité à basculer d'une tâche à l'autre comme :
- a) ☐ Très difficile
 - b) ☐ Difficile
 - c) ☐ Normale
 - d) ☐ Facile
 - e) ☐ Très facile

G - SATISFACTION

16. Comment avez vous trouvé l'interaction 3D avec le mode automatique ?
- a) ☐ Très difficile
 - b) ☐ Difficile
 - c) ☐ Normale
 - d) ☐ Facile
 - e) ☐ Très facile
17. Comment avez vous trouvé l'interaction 3D avec le mode manuel ?
- a) ☐ Très difficile
 - b) ☐ Difficile
 - c) ☐ Normale
 - d) ☐ Facile
 - e) ☐ Très facile
18. D'après-vous quelles améliorations seraient les bienvenues ?
19. Avez-vous des remarques à ajouter ?