



HAL
open science

Étude et mise en œuvre d'un système d'interaction adaptatif pour les applications de réalité virtuelle

M'Hamed Frad

► To cite this version:

M'Hamed Frad. Étude et mise en œuvre d'un système d'interaction adaptatif pour les applications de réalité virtuelle. Traitement du signal et de l'image [eess.SP]. Université Paris Saclay; Université d'Evry Val-d'Essonne; Ecole Nationale d'Ingénieurs de Monastir. Tunisia, 2016. Français. ⟨NNT : 2016SACLE053⟩. ⟨tel-01545845⟩

HAL Id: tel-01545845

<https://hal.science/tel-01545845v1>

Submitted on 23 Jun 2017

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.



HAL Authorization

NNT : 2016SACLE053



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PARIS-SACLAY
ET DE L'ÉCOLE NATIONALE D'INGÉNIEURS DE
MONASTIR
PRÉPARÉE À L'UNIVERSITÉ D'EVRY-VAL
D'ESSONNE

Ecole doctorale n°580

Sciences et Technologies de l'Information et de la Communication
Spécialité de doctorat : Robotique

par

M. MHAMED FRAD

Étude et mise en œuvre d'un système d'interaction adaptatif pour
les applications de réalité virtuelle

Thèse présentée et soutenue à Monastir, le 17 Décembre 2016.

Composition du Jury :

M. KAMEL BEN OTHMAN	Professeur ENIM	(Président du jury)
M. KUROSH MADANI	Professeur Université Paris-Est Créteil	(Rapporteur)
M. ANIS SAKLY	Maître de conférences ENIM	(Rapporteur)
M. SAMIR OTMANE	Professeur Université d'Evry-Val d'Essonne	(Examineur)
M. HICHEM MAAREF	Professeur Université d'Evry-Val d'Essonne	(Directeur de thèse)
M. ABDELLATIF MTIBAA	Professeur ENIM	(Directeur de thèse)

Titre : Étude et mise en œuvre d'un système d'interaction adaptatif pour les applications de réalité virtuelle

Keywords : interfaces d'interaction, tracking hybride, caractérisation, régression, réseaux de neurones, régression à vecteurs de support, calibration.

Résumé : Dans les dernières décennies, la réalité virtuelle a connu un essor fulgurant dans de nombreuses disciplines. Elle permet via des paradigmes d'interaction et d'immersion de plonger l'utilisateur au cœur d'un environnement artificiel créé numériquement. Ces paradigmes s'appuient sur l'utilisation des interfaces sensori-motrices bien spécifiques qui permettent à l'utilisateur d'interagir et accomplir des tâches particulières dans l'environnement virtuel. Néanmoins, des nombreux problèmes, d'origine technologique, sont souvent présents et peuvent pénaliser la qualité de l'interaction ainsi que le degré d'immersion de l'utilisateur dans l'environnement virtuel.

L'objectif de cette thèse est de proposer une procédure complète visant à guider l'utilisateur à calibrer une interface sensori-motrice spécifique et par conséquent tenter de pallier à certains défauts technologiques. L'originalité de la thèse réside dans l'utilisation d'une approche qui combine deux domaines de recherche qui ne s'associent que très rarement : celui du traitement de données et celui de la réalité virtuelle. Cette approche servira de cadre théorique et technique pour la conception d'une procédure de calibration complète permettant de garantir une interaction continue et précise dans l'environnement virtuel.

Afin de contrebalancer les défauts et limites techniques, le travail a été conduit sur plusieurs fronts : acquisition, traitement de données et validation. La première phase est marquée par l'utilisation d'un protocole innovant dans la mesure où il repose sur les techniques de réalité virtuelle pour récolter les données de calibration. Dans la deuxième phase, deux techniques de calibration ont été proposées pour améliorer la précision absolue de l'interface de réalité virtuelle. Les deux techniques se distinguent par leurs qualités d'approximateurs universels ainsi que par leurs capacités à estimer les sorties du système concerné à partir des entrées sans connaître à priori son modèle mathématique. Dans la dernière phase, deux prototypes d'applications de réalité virtuelle ont été développées pour s'assurer de la pertinence de notre approche.



Title : Design and Implementation of an adaptative interaction system for virtual reality applications

Keywords : interaction devices, hybrid tracking, characterization, regression, neural networks, support vector regression, calibration

Abstract : Over last decades, virtual reality has been widely used in many disciplines. It is able to plunge the user at the heart of an artificial environment created digitally through interaction and immersion paradigms. These paradigms are based on the use of very specific interfaces that help user to interact and perform specific tasks in the virtual environment. Nevertheless, many technical problems are often present and may penalize the quality of that interaction and may break user immersion in the virtual environment.

The goal of this thesis is to build a comprehensive procedure to guide the user to calibrate a virtual reality interface and therefore attempt to overcome some technological shortcomings. The originality of the thesis is the use of an approach that combines two areas of research that will combine very rarely, that of data processing and the virtual reality. This approach will provide theoretical and technical framework for the design of a comprehensive calibration procedure to ensure continuous and precise interaction in the virtual environment.

To overcome problems described above, the work was conducted on several fronts: data acquisition, processing and validation. The first step is by the use of a new protocol insofar as it is based on virtual reality techniques to collect calibration data. In second step, two calibration methods have been proposed to improve the absolute accuracy of the virtual reality interface. Both methods are universal approximators as well as their ability to estimate the outputs of the involved system from inputs even the model of the system being calibrated remains unknown. In the last step, two virtual reality applications prototypes were developed in order to assess the relevance of our approach.



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, Mr Abdellatif Mtibaa et Mr Hichem Maaref, pour avoir accepté d'être les directeurs de ma thèse, pour la confiance et la liberté de recherche qu'ils m'ont accordée ainsi que pour l'attention qu'ils portaient à mon travail.

Je remercie également Mr Samir Otmane pour son écoute et ses conseils ainsi que pour m'avoir aidé dans toutes les démarches scientifiques.

Je souhaite adresser des remerciements particulièrement intenses à ma chère mère Ramla, mon cher père Habib et à mon frère Anis. Ces quelques mots ne sauraient traduire ni tout ce que je vous dois ni mon admiration et encore moins ma profonde affection.

Un grand merci à ma chère épouse Inès, 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.

Mes cher amis Wahid, Mohamed, Khairallah, Hela, Amel, Sofène, Montassar et Zied ont bien évidemment droit, eux aussi, à mes remerciements pour tout leur soutien, tant dans les moments agréables que désagréables.

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 du laboratoire IBISC ainsi que du laboratoire EμE, qui sont maintenant devenus des amis et avec lesquels j'ai partagé tant de bons moments : merci Raja, Marwa, Cyrille, Alia, Siju et Ludovic ainsi que Pierre Boudoin.

Merci à tous.

Table des matières

Introduction Générale	1
1 La réalité virtuelle	5
Introduction	7
1.1 Evolution des différentes définitions, concepts et finalités	7
1.1.1 A l'origine de la réalité virtuelle	7
1.1.2 Définitions de la réalité virtuelle	7
1.2 Domaines en lien avec la Réalité Virtuelle	9
1.2.1 La réalité augmentée	9
1.2.2 La virtualité augmentée	10
1.3 Les domaines d'application de la réalité virtuelle	10
1.3.1 Application à la médecine	10
1.3.2 Application à l'industrie	12
1.3.3 Application à la science	13
1.4 Architecture d'un système de réalité virtuelle	14
1.5 Les interfaces matérielles de réalité virtuelle	15
1.5.1 Les dispositifs d'entrée	15
1.5.1.1 Les systèmes de tracking	15
1.5.1.1.1 Performances d'un système de tracking	16
1.5.1.1.2 Les technologies de tracking	18
1.5.1.1.2.1 Le tracking mécanique	18
1.5.1.1.2.2 Le tracking magnétique	18
1.5.1.1.2.3 Le tracking acoustique	19
1.5.1.1.2.4 Le tracking optique	20
1.5.1.1.2.4.1 Les caméras vidéo	20
1.5.1.1.2.4.2 Les caméras infrarouges	20
1.5.1.1.2.4.3 Les centrales inertielle	21
1.5.1.2 Les interfaces gestuelles	22
1.5.1.2.1 Les gants de données	22
1.5.1.2.2 La Wiimote	24
1.5.1.2.3 La Kinect	24
1.5.2 Les dispositifs de sortie	25
1.5.2.1 Les interfaces visuelles	25
1.5.2.2 Les interfaces à retour d'effort	28
1.5.2.2.1 Les interfaces fixes	30
1.5.2.2.1.1 Structure série	30
1.5.2.2.1.2 Structure parallèle	30
1.5.2.2.2 Les interfaces à câbles	32

1.5.2.2.3	Intégration des interfaces à retour d'effort au sein des environnements à grande échelle	33
1.5.2.3	Les interfaces auditives	34
Conclusion	35
2	Calibration d'un système de tracking en réalité virtuelle	37
Introduction	39
2.1	Les protocoles expérimentaux de calibration des systèmes de tracking	39
2.1.1	Supports physiques/virtuels pour l'acquisition des mesures de référence	40
2.1.2	Systèmes de tracking complémentaires pour l'acquisition des mesures de référence	44
2.2	Classification des protocoles de collecte de données	52
2.2.1	Collecte de données dynamique versus statique	52
2.2.2	Collecte de données automatique versus manuelle	52
2.3	Classification des erreurs	53
2.3.1	Les erreurs statiques	53
2.3.1.1	Distorsions statiques	53
2.3.1.2	Les erreurs aléatoires ou Gigue (Jitter)	53
2.3.2	Les erreurs dynamiques	54
2.3.3	L'erreur totale de tracking	55
2.4	Les techniques de calibration	55
2.4.1	L'interpolation tri-linéaire	55
2.4.2	L'interpolation par le point le plus proche	57
2.4.2.1	Méthode de pondération linéaire	57
2.4.2.2	Méthode de pondération exponentielle	57
2.4.2.3	Méthode de pondération basée sur le carré de la distance	58
2.4.2.4	Autres méthodes de pondération	58
2.4.3	Interpolation polynomiale	58
2.4.4	Les polynômes de Bernstein	60
2.4.5	Technique de Hardy HMQ (Hardy's multi-quadric method)	60
2.4.6	Technique basée sur les réseaux de neurones	61
2.5	Comparaison des techniques de calibration	63
Conclusion	64
3	Les techniques de régression non linéaire pour la calibration	65
Introduction	67
3.1	Les techniques de régression non linéaire	67
3.1.1	Les réseaux de neurones	67
3.1.1.1	Le neurone biologique	67
3.1.1.2	Les réseaux de neurones artificiels	68
3.1.1.2.1	Le neurone formel	69
3.1.1.2.2	Les réseaux de neurones artificiels	70
3.1.1.2.2.1	Les réseaux non bouclés	70
3.1.1.2.2.2	Les réseaux bouclés	71
3.1.1.2.3	L'apprentissage de réseaux de neurones artificiels	71
3.1.1.2.3.1	Apprentissage supervisé	71
3.1.1.2.3.2	Apprentissage non supervisé	72
3.1.1.3	Propriété fondamentale de la parcimonie des réseaux de neurones non bouclés à apprentissage supervisé	72

3.1.1.4	Le réseau perceptron multicouche MLP (Multi Layer Perceptron)	74
3.1.1.4.1	L'algorithme de rétropropagation du gradient de l'erreur	74
3.1.1.4.2	Configuration et optimisation des caractéristiques du réseau de neurones	77
3.1.1.4.2.1	Une, deux ou plusieurs couches cachées?	78
3.1.1.4.2.2	Nombre de neurones dans la couche cachée	78
3.1.1.4.2.3	Les méthodes d'apprentissage :	79
3.1.1.4.2.3.1	Apprentissage par lot	80
3.1.1.4.2.3.2	Apprentissage en ligne	80
3.1.1.4.2.3.3	Les méthodes d'optimisation	80
3.1.2	Régression à vecteurs de support	82
3.1.2.1	Formulation de la régression à vecteurs de support	82
3.1.2.2	Vue d'ensemble	88
3.1.2.2.1	Réglage des hyper paramètres	89
3.1.2.2.2	Démarche pour la mise en œuvre des modèles basés sur les machines à vecteurs de support	90
3.1.2.3	SVR et réseaux de neurones	91
Conclusion	92
4	Mise en place d'un protocole expérimental pour la caractérisation du Scalable-SPIDAR	93
Introduction	94
4.1	Plateforme expérimentale	94
4.2	Problèmes rencontrés dans le Scalable-SPIDAR	96
4.3	Caractérisation du Scalable-SPIDAR	98
4.3.1	L'initialisation multimodale du Scalable-SPIDAR	99
4.3.1.1	Paramètres multimodaux	100
4.3.1.2	Déroulement de l'initialisation du Scalable-SPIDAR	101
4.3.2	Protocole expérimental	102
4.3.2.1	Evaluation de l'ancien protocole	105
4.3.2.2	Protocole expérimental mis en place	106
4.3.2.3	Déroulement du nouveau protocole	109
4.3.2.4	Bilan de notre protocole de collecte de données	111
Conclusion	114
5	Caractérisation et calibration du Scalable-SPIDAR	115
Introduction	117
5.1	Hypothèses de travail (Volume utile)	117
5.2	Interprétation des résultats de caractérisation	118
5.2.1	Introduction	118
5.2.2	Incertitudes de position du Scalable-SPIDAR	118
5.2.3	Etude de la répétabilité	120
5.2.4	Comparaison avec les travaux de Pierre Boudoin	122
5.3	Calibration du Scalable-SPIDAR à l'aide des techniques de régression non linéaires	123
5.3.1	Calibration du Scalable-SPIDAR à l'aide des réseaux de neurones	124
5.3.1.1	Prétraitement des données	124
5.3.1.2	L'approche expérimentale	126

5.3.1.2.1	Configuration et optimisation des caractéristiques du réseau de neurones	126
5.3.1.2.2	Performances du réseau de neurones en apprentissage	128
5.3.1.2.3	Performances du réseau de neurones en généralisation	131
5.3.2	Calibration du Scalable-SPIDAR à l'aide des SVR	133
5.3.2.1	Performances des SVR en apprentissage	134
5.3.2.2	Performances des SVR en généralisation	135
5.3.3	Mise en balance des méthodes de calibration	136
5.4	Mise en pratique de la procédure de calibration du Scalable-SPIDAR	137
5.5	Applications pratiques	140
5.5.1	Modélisation 3D	140
5.5.1.1	Contraintes architecturales	140
5.5.1.2	Techniques de modélisation	140
5.5.1.3	Organisation des modèles 3D	142
5.5.2	L'Animation 3D	144
5.5.2.1	Les contraintes fonctionnelles	144
5.5.2.2	Techniques d'animation 3D	145
5.5.2.3	Squelettes des modèles 3D	146
5.5.3	Prototypes d'applications de réalité virtuelle	147
5.6	Les pistes de notre action	147
	Conclusion	149
	Conclusion Générale et Perspectives	151
	Références	153
	A Présentation de Virtools®	I
	B Programmation sous Virtools	IV

Liste des figures

1	Différents domaines en lien avec nos travaux de recherche	3
1.1	Le cube AIP : Autonomie, Interaction, Présence, ZELTZER [1992]	8
1.2	Le continuum réalité virtualité, MILGRAM et collab. [1994]	9
1.3	Un modèle virtuel similaire aux organes du patient pour l'entraînement aux actes chirurgicaux, [CRAIG et collab., 2009].	10
1.4	La réalité virtuelle au secours des enfants autistes, CRAIG et collab. [2009]	11
1.5	Le professeur Henry Fuchs assiste le docteur Etta Pisano lors d'une simple visualisation à l'aide d'un système de réalité augmentée développé à l'UNC, CRAIG et collab. [2009]	12
1.6	Représentation virtuelle de l'intérieur du sein, CRAIG et collab. [2009]	12
1.7	Système de Prototypage Virtuel de Caterpillar, CRAIG et collab. [2009]	13
1.8	Étude de l'habitable d'une voiture dans un CAVE, ZIMMERMANN [2008]	13
1.9	Un robot en train d'explorer les profondeurs de l'océan, CRAIG et collab. [2009].	14
1.10	Architecture d'un système de réalité virtuelle, COIFFET [1996]	14
1.11	Précision, résolution et répétabilité d'un système de tracking, [BURDEA et COIFFET, 2003]	16
1.12	Gigue d'un système de tracking, [BURDEA et COIFFET, 2003]	16
1.13	Latence d'un système de tracking, [BURDEA et COIFFET, 2003]	17
1.14	Dérive d'un système de tracking, [BURDEA et COIFFET, 2003]	17
1.15	La fréquence d'échantillonnage d'un système de tracking en fonction du nombre de récepteurs, [BURDEA et COIFFET, 2003]	18
1.16	Le BOOM (Binocular Omni-Orientation Monitor) , [FAKESPACE, 1994]	19
1.17	Système Aurora, FISCHER [2005].	19
1.18	Le système de tracking acoustique IS-900, [INTERSENSE, 1999]	20
1.19	Le principe fonctionnement du système optique, ART ADVANCED REALTIME TRACKING [2016]	21
1.20	La centrale inertielle MTx , XSSENS [2016]	22
1.21	Le Data Glove Ultra, FIFTH DIMENSION TECHNOLOGIES [2016]	22
1.22	Le CyberGlove, CYBER GLOVES SYSTEMS [2016]	23
1.23	Le CyberGrasp, CYBER GLOVES SYSTEMS [2016]	23
1.24	Le CyberTouch, CYBER GLOVES SYSTEMS [2016]	23
1.25	La Wiimote, NINTENDO [2010]	24
1.26	Les éléments de la Kinect, MICROSOFT [2016]	25
1.27	Un CAVE à 5 écrans de projection, IGATECHNOLOGIES [2016]	26
1.28	Interface visuelle de réalité virtuelle, le C6. Des paires d'images sont rétro-projetées sur les faces du cube permettant ainsi une vision stéréoscopique omnidirectionnelle, VIRTUAL REALITY APPLICATIONS CENTER OF IOWA STATE UNIVERSITY VRAC [2016]	27

1.29 Le SAS cube, CLARTE [2016]	27
1.30 Un visiocasque de Nvis avec accusation optique de la position de la tête, Nvis [2016]	28
1.31 Les casques de Nvis, Nvis [2016]	28
1.32 Système maître-esclave pour la téléopération, GOSSELIN [2005]	29
1.33 Interfaces à retour d'effort à réaction interne et externe, FUCHS et collab. [2011]	29
1.34 PHANToM Desktop, GEOMAGIC® HAPTIC DEVICES [2016].	30
1.35 Virtuouse, HAPTION [2016].	31
1.36 L'interface SHaDe, BIRGLEN et collab. [2002]	31
1.37 Le cobot haptique, FAULRING et collab. [2004]	32
1.38 L'interface à retour d'effort SPIDAR-8, WALAIRACHT et collab. [2002]	33
1.39 L'interface à retour d'effort SPIDAR-H, HASEGAWA et collab. [2006]	33
1.40 Comparaison sur l'intégration des interfaces à retour d'effort dans les es- paces à grandes dimensions , KUNZLER et RUNDE [2005]	34
2.1 Exemple d'un support physique typique	40
2.2 Equipements utilisés pour l'acquisition de la mire de calibration, KINDRA- TENKO et SHERMAN [2005].	41
2.3 Schéma de la plateforme expérimentale, ZACHMANN [2000]	42
2.4 Schéma des principaux éléments de montage expérimental , DAY et collab. [1998]	43
2.5 Le volume de calibration, OCHOA-MAYORGA et collab. [2005].	43
2.6 La plateforme de mesure en polycarbonate introduite par HUMMEL et col- lab. [2005].	44
2.7 Protocole volumétrique pour la calibration des systèmes de tracking élec- tromagnétiques, HIMBERG et collab. [2012]	44
2.8 Le curseur est placé dans chacun des cubes, GHAZISAEDY et collab. [1995].	45
2.9 Les points collectés sont décalés d'une distance par rapport au centre du cube, GHAZISAEDY et collab. [1995].	45
2.10 Montage expérimental pour la collecte des mesures utilisant un bras de FARO, LIVINGSTON et STATE [1997].	46
2.11 La procédure de ré-échantillonnage.	47
2.12 Le stylet de mesure hybride utilisé dans l'expérimentation, BIRKFELLNER et collab. [1998].	47
2.13 Salle de chirurgie utilisée pour les expérimentations, BIRKFELLNER et collab. [1998].	48
2.14 Plateforme flexible utilisée pour fixer les capteurs et les marqueurs optiques, IKITS et collab. [2001].	48
2.15 Montage de calibration : la mire est composée de 40 trous espacés de 20cm , PÉRIÉ et collab. [2002].	49
2.16 Montage utilisé dans le protocole de calibration volumique , FRANTZ et col- lab. [2003].	50
2.17 Un robot Lego en plastique pour la collecte de mesures , WU et TAYLOR [2003].	50
2.18 Exemples d'implants et prothèses dentaires, MOREIRA et collab. [2014].	51
2.19 Configuration du système : (A) L'objet, (B) Le système de tracking magné- tique Polhemus et (C) le robot ABB IRB 140, MOREIRA et collab. [2014].	51
2.20 Classification des erreurs de tracking.	53
2.21 Retard de synchronisation (Synchronisation delay).	54

2.22	L'interpolation tri linéaire.	56
2.23	Comparaison des résultats obtenus pour la position (a) et l'orientation (b). Calibration à l'aide un polynôme d'ordre 4, interpolation par le plus proche voisin et calibration basée sur les réseaux de neurones, KINDRATENKO et SHERMAN [2005]	62
2.24	L'erreur moyenne de position après la calibration en fonction du nombre de neurones, KINDRATENKO et SHERMAN [2005]	62
3.1	Schéma de principe d'un neurone biologique	68
3.2	Principe d'un neurone formel	69
3.3	Un réseau de neurones non bouclé à n entrées, une couche de N_c neurones cachés et N_s neurones de sortie	70
3.4	Réseau de neurones à $n + 1$ entrées, N_c neurones cachés à fonction d'activa- tion sigmoïde et un neurone de sortie linéaire	71
3.5	Un réseau de neurones bouclé. Les nombres dans les carrés sont les retards associés à chaque connexion	72
3.6	Réseau MLP à une seule couche cachée	75
3.7	Bon ajustement, Mauvais ajustement	77
3.8	Minimisation de la fonction coût par la méthode de gradient simple, DREY- FUS et collab. [2008]	81
3.9	La fonction de perte de Vapnik ϵ -sensible	84
3.10	Deux approximations linéaires à l'intérieur du tube ayant le même risque empirique R_{emp}^e sur les données d'apprentissage que la fonction de régres- sion. Les carreaux remplis sont des vecteurs de support alors que les vides ne le sont pas.	84
3.11	Architecture de la méthode de régression basée sur les SVM	88
3.12	Démarche de développment d'un modèle basée sur les SVMs	91
4.1	Disposition de plateforme expérimentale	96
4.2	Le moteur et la poulie sur son axe	98
4.3	Vue détaillée d'un moteur du Scalable-SPIDAR et du guide d'enroulement	99
4.4	Interface de communication avec le Scalable-SPIDAR (boîte noire)	99
4.5	Régions d'initialisation et cadre expérimental	101
4.6	Organigramme de la procédure d'initialisation du Scalable-SPIDAR	103
4.7	Correspondance du référentiel du tracking optique ARTtrack1 avec celui du Scalable-SPIDAR	104
4.8	Référentiel du système de tracking optique ARTtrack1	104
4.9	Déroulement de l'ancien protocole. A gauche, un utilisateur est en train d'effectuer la saisie des positions. A droite, une illustration de ce que per- çoit l'utilisateur	105
4.10	Architecture du protocole proposé	106
4.11	Le module Calculate IC	108
4.12	Le module collision detection	108
4.13	Le module DCCU Tracker	109
4.14	Le module DCCU Init	109
4.15	Cadre expérimental du nouveau protocole. Un guide visuel d'aide à la sélec- tion indique à l'utilisateur le cube à saisir.	110
4.16	Représentation virtuelle de l'espace de travail : Chaque cube est muni d'une sphère autour de son centre.	110
4.17	Ancien protocole de collecte de données	112

4.18	Nouveau protocole de collecte de données	112
4.19	Représentation virtuelle idéale de notre espace de travail	113
4.20	Reconstruction de notre espace de travail à partir des mesures relevées lors d'une campagne de mesure	113
5.1	Représentation du volume utile en bleu du Scalable-SPIDAR à l'intérieur de sa structure (en noir).	117
5.2	Répartition de l'erreur absolue de position actuelle dans l'espace de travail	119
5.3	Erreur absolue représentée en séquence	119
5.4	Diagramme en boîte de l'erreur de position	120
5.5	Histogramme de la distribution d'erreur absolue de position	121
5.6	Etude de l'erreur de répétabilité	121
5.7	Répartition de l'erreur absolue de position dans l'espace de travail relevée par Pierre Boudoin	122
5.8	Chaîne de prétraitement de données	124
5.9	Décomposition des données récoltées en trois sous ensembles : apprentis- sage, validation et test	126
5.10	Architecture du réseau de neurones de base utilisé pour la calibration du Scalable-SPIDAR	127
5.11	Erreur absolue moyenne en fonction du nombre de neurones dans la couche cachée	128
5.12	Performance du réseau de neurones employé pour la calibration	128
5.13	Régression en fonction des sous ensembles des données utilisées	129
5.14	Erreur absolue de position du Scalable-SPIDAR par rapport à la distance du centre avant et après calibration par réseau de neurones	130
5.15	Représentation spatiale de l'erreur absolue après calibration par le réseau de neurones	131
5.16	Erreur absolue de position du Scalable-SPIDAR avant et après calibration à l'aide du réseau de neurones	132
5.17	Erreur absolue de position du Scalable-SPIDAR avant et après généralisation	133
5.18	Erreur absolue de position du Scalable-SPIDAR par rapport à la distance du centre avant et après calibration par SVR	134
5.19	Représentation spatiale de l'erreur absolue après calibration par les SVR	135
5.20	Erreur absolue de position du Scalable-SPIDAR avant et après calibration à l'aide des SVR	135
5.21	Erreur absolue de position du Scalable-SPIDAR avant et après généralisa- tion à l'aide des SVR	136
5.22	Comparaison entre les performances de deux méthodes de calibration ré- seaux de neurones et SVR en généralisation	137
5.23	L'unité de collecte et de correction de données	138
5.24	Boucle de traitement de données	139
5.25	Dimensions de la station robotisée ACRS	141
5.26	Exemple des opérateurs booléens.	141
5.27	Exemples de maillage.	142
5.28	Modèle 3D du Scalable-SPIDAR	142
5.29	Modèle 3D de l'effecteur du Scalable-SPIDAR	143
5.30	Modèles 3D de la base de la station robotisée et sa roue	143
5.31	Modèle 3D du magasin de la station ACRS	144
5.32	Modélisation 3D de l'articulateur du robot	144

5.33	Modèle 3D du chariot des pièces	144
5.34	Modèle 3D complet de la station ACRS	145
5.35	Squelette de l'effecteur du Scalable-SPIDAR	146
5.36	Squelette de la station ACRS	146
5.37	Ecart entre la position de référence et la position mesurée par Scalable- SPIDAR avant calibration	147
5.38	Ecart entre la position de référence et la position obtenue après calibration	148
5.39	Méthode de développement à l'aide du Système générateur de Xilinx (XGS)	148
A.1	Représentation d'un Building Block.	I
A.2	Le schematic de Virtools®	II
B.1	Les différents scripts chargés dans Virtools	IV

Liste des tableaux

1.1	Caractéristiques des interfaces SPIDAR I et SPIDAR II	32
2.1	Résumé des différentes techniques de calibration, KINDRATENKO [2000], KINDRATENKO et SHERMAN [2005]	63
5.1	Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR	122
5.2	Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après calibration par le réseau de neurones	131
5.3	Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après généralisation	132
5.4	Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après calibration à l'aide des SVR	134
5.5	Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après généralisation à l'aide des SVR	136
5.6	Comparaison des valeurs caractéristiques des erreurs absolues de deux méthodes en généralisation	137

Introduction Générale

La réalité virtuelle suscite sans doute beaucoup d'intérêt dans de nombreux domaines tels que la médecine, l'industrie de l'automobile et les arts numériques et interactifs. Elle constitue un outil d'aide à la conception, de la visualisation des données scientifiques ainsi que de la simulation de situations d'urgence, sportives et d'entraînement (conduite des véhicules, avions ...). La réalité virtuelle permet au travers des paradigmes d'interaction et d'immersion de plonger un ou plusieurs utilisateurs au cœur d'un environnement virtuel dans lequel il peut ressentir et interagir en temps réel. Derrière le côté attrayant de ces paradigmes se cachent différentes familles d'interfaces sensori-motrices qui permettent à l'utilisateur d'interagir et d'accomplir des tâches dans l'environnement virtuel. Néanmoins, ces interfaces sensori-motrices sont souvent sujettes à des problèmes, dus essentiellement aux différentes technologies utilisées et qui peuvent entraîner un comportement erratique du système de réalité virtuelle. Ces problèmes peuvent perturber l'expérience ainsi le degré d'immersion de l'utilisateur dans l'environnement virtuel.

Cette thèse se propose, pour répondre à ces différents besoins et questions, une approche combinant les recherches en réalité virtuelle et en traitement de données. La réalité virtuelle assure la prise en compte des interfaces d'interaction. Le traitement de données, quant-à lui, offre des pistes pour la gestion des données issues de ces interfaces. L'identification des interfaces sensori-motrices d'interaction servira de cadre théorique et technique pour l'introduction des protocoles et méthodes de calibration. Une fois le cadre théorique posé, nous souhaitons proposer des solutions dans le but de tendre vers une gestion de données continue et précise provenant des interfaces d'interaction lors d'une expérience de réalité virtuelle.

Dans les systèmes de réalité virtuelle, l'utilisateur est immergé dans un environnement composé de plusieurs interfaces sensori-motrices pour offrir à l'utilisateur la possibilité d'accomplir une tâche particulière. Un tel système s'apparente bien à un système multi-capteurs dont les objectifs sont nombreux :

- * Remédier aux limitations technologiques : Chaque interface sensori-motrice est composée d'un ou plusieurs capteurs afin de permettre à l'utilisateur d'interagir avec l'environnement virtuel. Bien entendu chacun des capteurs est basé sur une technologie qui lui permet de récupérer un certain nombre de caractéristiques d'un phénomène physique. Ceci peut souvent entraîner des erreurs ou une mauvaise interprétation des données en provenance du capteur.
- * Créer de l'information : Chaque capteur est capable de récupérer un certain type de données. En combinant différents capteurs et donc différentes sources de données, nous pouvons créer une nouvelle information en s'appuyant sur les techniques de fusion de données.

Pour répondre à cette problématique, ce travail propose de broser les grandes lignes de la gestion des interfaces d'interaction et de leurs données.

Les contributions de cette thèse s'articulent autour de l'objectif suivant : pour tendre à un système d'interaction performant en réalité virtuelle, nous devons proposer une procédure de calibration suffisamment crédible qui garantit la qualité et la continuité de l'interaction et par conséquent l'expérience vécue par l'utilisateur au cœur l'environnement virtuel.

La première contribution est de présenter les fondements théoriques et techniques de la réalité virtuelle. Nous présentons donc les définitions clés de la réalité virtuelle ainsi que ses différents domaines d'applications. Nous identifions également les différentes interfaces matérielles rencontrées en réalité virtuelle en pointant du doigt les principaux défauts et qualités de chacun. Ceux-ci auront été l'occasion de développer le cadre général de notre travail mais aussi de déceler les problèmes inhérents aux interfaces de réalité virtuelle en particulier le problème de la calibration.

La deuxième contribution est une analyse bibliographique approfondie des différentes méthodes d'acquisition et de traitement de données utilisées dans le processus de calibration des systèmes de "tracking" en réalité virtuelle. Cette analyse soulève un point très important : il n'existe pas une méthode générique ou universelle pour calibrer un système de tracking. Pour cela, nous nous sommes attachés à établir des comparaisons entre les différentes méthodes identifiées dans la littérature afin de tendre vers la meilleure calibration possible. La relation entre la réalité virtuelle et les protocoles et les techniques de calibration est par ailleurs particulière. Nous dégageons les bénéfices croisées entre ces domaines et nous voyons en quoi la réalité virtuelle et les méthodes de traitement de données s'auto alimentent.

La troisième contribution consiste à l'exploration et la proposition d'une procédure de calibration complète d'une interface à retour d'effort à échelle humaine peu précise : le Scalable-SPIDAR (SPace Interface Device for Artificial Reality), par rapport à un autre système plus précis, le tracking optique afin de garantir la précision et la continuité des données. L'originalité de la procédure réside dans la conception d'un protocole expérimental innovant dans la mesure où il s'appuie sur les techniques de réalité virtuelle pour l'acquisition des données de calibration. De plus, nous proposons d'explorer les méthodes de régression non linéaires pour améliorer les performances de l'interface en question. En effet, meilleures sont les performances des interfaces plus forte sera la qualité et la continuité de l'interaction dans l'environnement virtuel.

La quatrième contribution débouche sur la mise en œuvre de deux prototypes d'application de réalité virtuelle dans le but de valider la pertinence de la procédure de calibration. Une fois la méthode de calibration choisie, nous mettons en pratique la procédure de calibration. Nous proposons alors une unité de collecte et correction de données permettant d'agglomérer les différentes interfaces sous forme d'interfaces virtuelles. Celles-ci bénéficient des avantages de méthodes de traitement de données. Les prototypes serviront à évaluer les performances des méthodes de calibration et à montrer l'intérêt d'une approche combinant les recherches en réalité virtuelle et les méthodes de traitement de données.

La thèse s'organise autour du triptyque suivant : réalité virtuelle, protocoles et techniques de calibration. La figure 1 annonce le plan de la thèse.

Dans le premier chapitre, nous présentons les différentes définitions de réalité virtuelle avancées par la communauté. Nous montrons également son utilité via ses multiples domaines d'application. Nous nous intéressons ensuite aux interfaces sensori-motrices rencontrées en réalité virtuelle. Nous expliciterons alors les défauts et qualités des

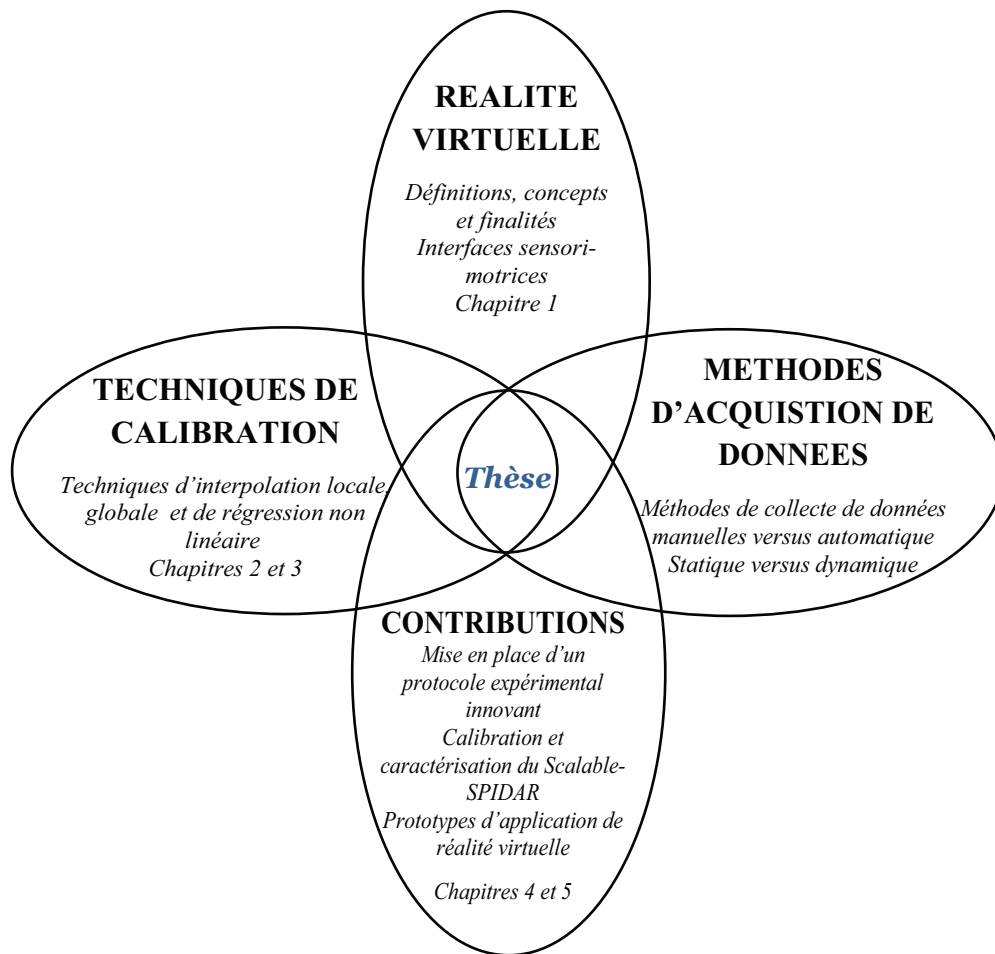


FIGURE 1 – Différents domaines en lien avec nos travaux de recherche

technologies utilisées et les problématiques qui en découlent.

Le second chapitre est consacré à l'exploration des méthodes utilisées pour la calibration des systèmes de "tracking" en réalité virtuelle. Nous nous proposons de revenir sur les protocoles expérimentaux permettant l'acquisition des données de calibration. Nous présentons ensuite les techniques proposées pour corriger les différentes erreurs de tracking.

Le troisième chapitre présente une synthèse théorique et pratique de deux techniques de régression non linéaires et d'approximateurs universels. Nous discuterons précisément les réseaux de neurones et la régression à vecteurs de support. Nous verrons en outre qu'en l'absence d'un modèle analytique du système de tracking une régression peut être utilisée pour la calibration.

Le quatrième chapitre traite de la caractérisation d'une interface haptique disponible mais peu précise par rapport à une autre interface plus précise et sûre qui est souvent sujette à occultation. Dans notre étude, ceci correspond à calibrer un Scalable-SPIDAR qui est une interface à retour d'effort à échelle humaine permettant entre autre le tracking de la main par rapport à un système de tracking optique dans le but d'améliorer sa précision en position et par conséquent garantir la continuité des données. Dans un premier temps, une présentation de la plateforme expérimentale est menée. Ensuite, une étude succincte

sur les problèmes inhérents à la conception de Scalable-SPIDAR est réalisée afin de le caractériser. Puis un protocole expérimental basé sur les techniques de réalité virtuelle est mis en place pour récolter les données de calibration.

Le cinquième et dernier chapitre propose une approche de calibration type boîte noire, basée sur les deux techniques de régression non linéaire abordées déjà et d'ores dans le chapitre 3 (réseaux de neurones et régression à vecteurs de support) pour calibrer le Scalable-SPIDAR. Ensuite, une unité de traitement de données en temps réel est conçue pour pouvoir traiter et corriger les données issues des interfaces d'interaction de réalité virtuelle à la volée. Puis, nous présentons deux prototypes d'applications virtuelles conçues pour évaluer la pertinence et l'intérêt de la procédure de calibration. Nous clôturons la thèse par une conclusion générale et des perspectives.

Chapitre 1

La réalité virtuelle

Résumé

Ce premier chapitre est consacré à la présentation du domaine de réalité virtuelle. La première section propose de revenir sur quelques définitions de la réalité virtuelle avancées par la communauté. Dans une deuxième section, nous décrivons les principaux domaines d'applications de la réalité virtuelle. Puis, nous présentons les différentes interfaces matérielles rencontrées en réalité virtuelle. Nous explicitons alors les avantages et inconvénients des technologies utilisées et les problématiques qui en découlent.

Sommaire

Introduction	7
1.1 Evolution des différentes définitions, concepts et finalités	7
1.1.1 A l'origine de la réalité virtuelle	7
1.1.2 Définitions de la réalité virtuelle	7
1.2 Domaines en lien avec la Réalité Virtuelle	9
1.2.1 La réalité augmentée	9
1.2.2 La virtualité augmentée	10
1.3 Les domaines d'application de la réalité virtuelle	10
1.3.1 Application à la médecine	10
1.3.2 Application à l'industrie	12
1.3.3 Application à la science	13
1.4 Architecture d'un système de réalité virtuelle	14
1.5 Les interfaces matérielles de réalité virtuelle	15
1.5.1 Les dispositifs d'entrée	15
1.5.1.1 Les systèmes de tracking	15
1.5.1.1.1 Performances d'un système de tracking	16
1.5.1.1.2 Les technologies de tracking	18
1.5.1.1.2.1 Le tracking mécanique	18
1.5.1.1.2.2 Le tracking magnétique	18
1.5.1.1.2.3 Le tracking acoustique	19
1.5.1.1.2.4 Le tracking optique	20
1.5.1.2 Les interfaces gestuelles	22

1.5.1.2.1	Les gants de données	22
1.5.1.2.2	La Wiimote	24
1.5.1.2.3	La Kinect	24
1.5.2	Les dispositifs de sortie	25
1.5.2.1	Les interfaces visuelles	25
1.5.2.2	Les interfaces à retour d'effort	28
1.5.2.2.1	Les interfaces fixes	30
1.5.2.2.1.1	Structure série	30
1.5.2.2.1.2	Structure parallèle	30
1.5.2.2.2	Les interfaces à câbles	32
1.5.2.2.3	Intégration des interfaces à retour d'effort au sein des environnements à grande échelle	33
1.5.2.3	Les interfaces auditives	34
Conclusion		35

INTRODUCTION

Devant la profusion des définitions autour de l'expression "réalité virtuelle" qui, depuis des années, apparaît clairement dans les médias tantôt pour présenter de nouveaux jeux vidéo et tantôt pour louer le réalisme des effets spéciaux dans les films d'animation, nous pouvons constater que cette expression possède un périmètre de définition ample. Cette section revient au préalable sur les origines de la réalité virtuelle. Nous nous proposons d'en préciser les différents concepts de base présents dans la littérature.

1.1 EVOLUTION DES DIFFÉRENTES DÉFINITIONS, CONCEPTS ET FINALITÉS

Bien que dans nombreux domaines scientifiques la définition du domaine de recherche soit claire, la réalité virtuelle a la particularité que sa simple définition est sujette à débat.

1.1.1 A l'origine de la réalité virtuelle

L'expression "Réalité Virtuelle" est une traduction littérale de l'expression Virtual Reality attribuée à Jaron Lanier. Fondateur de la société VPL (Visual Programming Language Research) qui commercialisait les premiers équipements complets de réalité virtuelle. C'est en juin 1989 en cherchant une accroche marketing lors d'une démonstration d'une version interactive d'Alice au Pays de Merveilles, qu'il inventa l'expression réalité virtuelle **KEVIN et collab. [1989]**. Cette expression allie deux termes réalité et virtuelle des sens opposés. C'est un bel oxymoron qui n'aura pas facilité la compréhension de la réalité virtuelle par le grand public. Jean-Paul Papin dans le traité de la réalité virtuelle **FUCHS et MOREAU [2003]** rappelle que cette traduction directe ne rend pas compte de signification exacte du mot "Virtual" qui signifie plutôt "pratiquement" et lui préfère l'expression de réalité vicariante ou encore environnement vicariant. **ELLIS [1991]** quant à lui préfère l'expression "d'environnement virtuel" qui offre l'avantage d'être sémantiquement plus cohérente que l'expression réalité virtuelle.

1.1.2 Définitions de la réalité virtuelle

La littérature regorge de définitions qui, bien que souvent différentes, se retrouvent sur un certain nombre de points clés. Nous ne présentons ici qu'un échantillon que nous espérons représentatif. Précisons que nous excluons les définitions qui limitent la réalité virtuelle à des images de synthèse ou à l'emploi d'une interface spécifique.

BURDEA et COIFFET [1993] définissent la réalité virtuelle autour des trois composantes : Immersion, Interaction et Imagination. En effet, 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.

ELLIS [1991] propose la définition suivante :

"Les environnements virtuels sont des images virtuelles interactives améliorées par des processus particuliers et complétées par des restitutions provenant d'autres modalités telles

que le son et l'haptique et ce pour convaincre les utilisateurs qu'ils sont immergés dans un espace synthétique".

ZELTZER [1992] propose un triptyque constitué de l'autonomie, de l'interaction et de la présence. L'autonomie représente le comportement des éléments virtuels présents dans l'environnement virtuel. L'interaction désigne la capacité d'interagir avec ces éléments et l'environnement. La présence est ici liée à la stimulation sensorielle. Autonomie, interaction et présence sont considérées comme les dimensions caractérisant l'espace des environnements virtuels (figure 1.1 pour une représentation de cet espace).

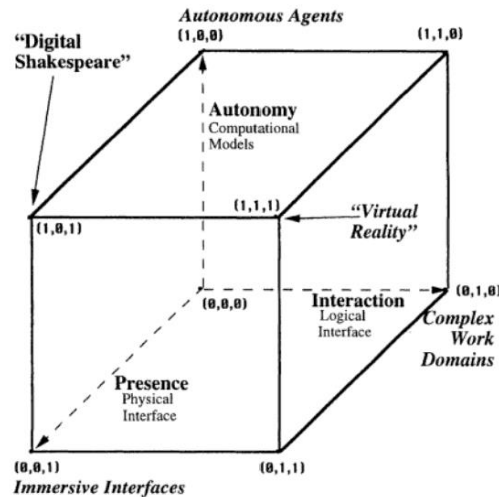


FIGURE 1.1 – Le cube AIP : Autonomie, Interaction, Présence, ZELTZER [1992]

En 2000, Jaron Lanier, inventeur de l'expression Virtual Reality, retouche sa vision de la réalité virtuelle dans une interview MCKENNA [2000].

"La réalité virtuelle permet via l'informatique de créer l'illusion d'être dans un monde alternatif avec d'autres personnes. Il s'agit d'une sorte de rêve que vous faites de façon consciente et auquel d'autres personnes participent".

Finalement, la communauté française s'est concertée autour d'une finalité et deux définitions. Voici comment FUCHS et MOREAU [2003] définissent 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."

FUCHS et MOREAU [2003] proposent également une définition fonctionnelle de la réalité virtuelle :

"La réalité virtuelle va permettre à l'homme de s'extraire de la réalité physique pour changer virtuellement de temps, de lieu et(ou) de type d'interaction : interaction avec un environnement simulant la réalité ou interaction avec un monde imaginaire ou symbolique".

ARNALDI et collab. [2003] suggèrent une définition technique :

"La réalité virtuelle est un domaine scientifique et technique exploitant l'informatique et des interfaces comportementales en vue de simuler dans un monde virtuel le comportement d'entités 3D, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion pseudo-naturelle par l'intermédiaire de canaux sensori-moteurs."

La définition technique permet de préciser avec efficacité les fonctionnalités d'un système de réalité virtuelle. L'énoncé de la finalité et la définition fonctionnelle, quant-à elles, sont centrées plus sur l'expérience dans le monde artificiel synthétisé.

1.2 DOMAINES EN LIEN AVEC LA RÉALITÉ VIRTUELLE

Nous n'avons pas parlé jusqu'ici d'un domaine de recherche voisin de la réalité virtuelle : la réalité mixte. Contrairement aux techniques de réalité virtuelle qui, pour mieux faire immerger l'utilisateur dans l'environnement virtuel, vont tenter de le couper sensoriellement de l'environnement physique, la réalité mixte va elle fondre environnement physique et entités virtuelles.

MILGRAM et collab. [1994] ont proposé un continuum allant du réel au virtuel avec la part de chacun des deux mondes. Ils définissent ainsi la réalité mixte qui occupe l'espace entre les deux extrêmes et réunit deux sous domaines : la réalité augmentée et la virtualité augmentée.

La figure 1.2 décrit une représentation de ce continuum.

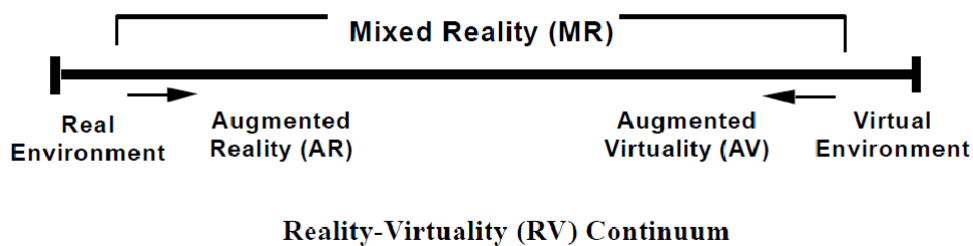


FIGURE 1.2 – Le continuum réalité virtualité, MILGRAM et collab. [1994]

La réalité augmentée propose un contenu à forte proportion réelle alors que la virtualité augmentée sera à forte proportion virtuelle.

1.2.1 La réalité augmentée

La réalité augmentée rajoute des informations virtuelles au-dessus de l'environnement réel. Les informations virtuelles ajoutées sont essentiellement visuelles et consistent en des images de synthèse créées par un ordinateur et rendues visibles à l'utilisateur par le biais d'un dispositif tel qu'un écran d'ordinateur, un visiocasque ou tout autre dispositif de projection.

1.2.2 La virtualité augmentée

La virtualité augmentée incorpore à l'environnement virtuel des entités physiques réelles. Ces entités réelles intègrent donc l'environnement virtuel et peuvent donc interagir avec celui-ci.

1.3 LES DOMAINES D'APPLICATION DE LA RÉALITÉ VIRTUELLE

La section précédente nous a conduit à définir la réalité virtuelle, il est temps maintenant de souligner son utilité via ses différents domaines d'application. Ces dernières années, la réalité virtuelle a suscité sans doute un vif intérêt dans nombreux et différents domaines. La médecine, l'industrie et la science sont des secteurs d'activités différents mais qui font tous appel aux technologies de la réalité virtuelle.

1.3.1 Application à la médecine

La médecine, l'un de principaux domaines d'application de la réalité virtuelle, cherche sans cesse à innover. Elle s'intéresse de plus en plus aux systèmes dits de réalité virtuelle qui semblent offrir des possibilités d'efficacité dans la formation de personnel médical. Par exemple, le système BDI (Boston Dynamic INc) est un simulateur pour les enseignements de la chirurgie qui utilise divers moyens informatiques et mécaniques pour reproduire les sensations visuelles et gestuelles d'un chirurgien lors d'une opération chirurgicale. La figure 1.3 montre le principe d'un simulateur chirurgical qui se compose de trois unités : la première est l'unité d'affichage, la deuxième unité est le modèle déformable représentant les caractéristiques géométriques et le comportement biomécanique à simuler et la troisième unité est le système d'interaction avec les organes déformables. Cette interaction est assurée par l'utilisation des périphériques à retour d'effort. Nous aborderons ce point particulier dans la prochaine partie dédiée aux interfaces de réalité virtuelle.



FIGURE 1.3 – Un modèle virtuel similaire aux organes du patient pour l'entraînement aux actes chirurgicaux, [CRAIG et collab., 2009].

La réalité virtuelle, par sa qualité multi-sensorialité et ses capacités d'interaction, peut contribuer à la remédiation cognitive. En effet, les technologies de la réalité virtuelle permettent d'évaluer les capacités cognitives et logiques des personnes à travers des exercices

et de les faire travailler sur la boucle perception-action dans le but de mettre en place des démarches de remédiation. Plusieurs projets explorent ce domaine d'application. **STRICKLAND [1996]**, porteur de projet réalisé conjointement entre l'université de la Caroline de Nord et le Département de Traitement et Education des enfants autistes ou atteints de troubles de la communication associée (TEACCH), s'est penché sur le développement des environnements virtuels immersifs adaptés pour les enfants autistes. Les enfants sont immergés dans un environnement virtuel simple au moyen des visiocasques dans lesquels ils accomplissent des tâches. Les enfants apprenaient dans un premier temps à reconnaître l'objet (voiture par exemple) puis à l'identifier dans une scène qui se compose des éléments simples (voitures + bâtiments). Ce genre d'expérience a permis de mesurer les capacités des enfants autistes et de réévaluer les performances (figure 1.4).

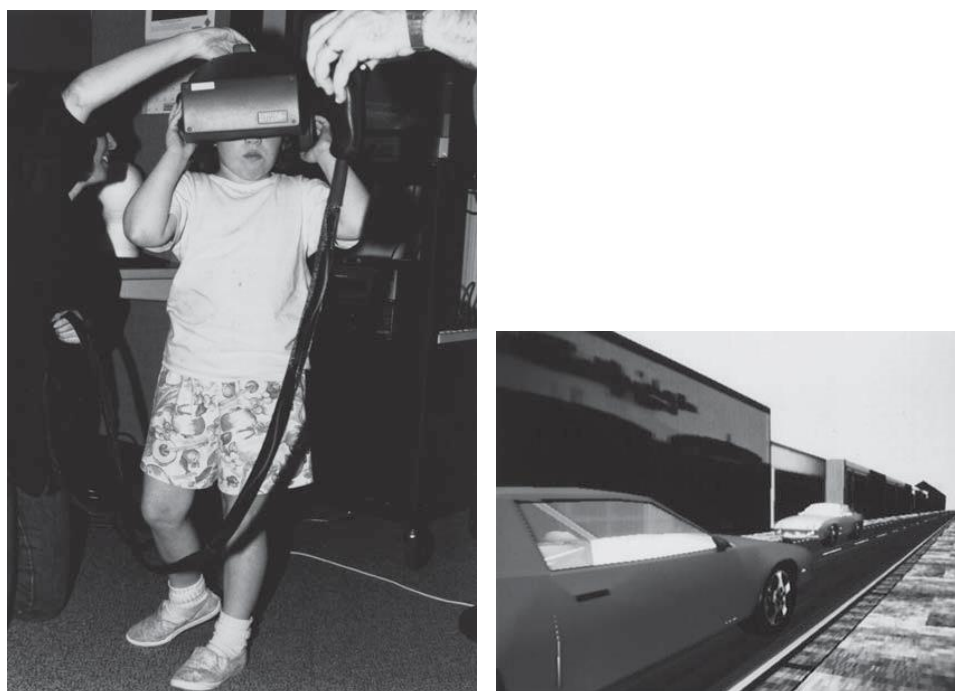


FIGURE 1.4 – La réalité virtuelle au secours des enfants autistes, **CRAIG et collab. [2009]**

Pour améliorer les outils d'interaction et de perception du personnel médical avec le monde réel, d'autres travaux se sont appuyés sur les technologies de réalité augmentée. Le personnel est alors immergé dans un environnement réel/virtuel plus riche en informations. C'est en 1992, que le professeur Henry Fuchs initia un projet de réalité augmentée à l'université de Caroline du Nord qui a permis de mettre au point un système de diagnostic à ultrasons (figure 1.5).

Comme première expérimentation, il a montré les résultats d'une échographie obstétricale. La visualisation est effectuée en utilisant une caméra montée sur le visiocasque de l'opérateur. Les images de la caméra et les images ultrasonores générées par ordinateur sont superposées sur le ventre de la mère permettant au gynécologue d'examiner la position du fœtus dans l'utérus de la mère **BAJURA et collab. [1992]**. La réalité augmentée pourrait être également très utile dans la chirurgie. Elle permet de guider la précision de la tâche. Par exemple, comment exécuter une biopsie d'aiguille d'une petite tumeur du sein, où les objets virtuels identifient l'endroit de la tumeur et guident l'aiguille à sa cible (figure 1.6).



FIGURE 1.5 – Le professeur Henry Fuchs assiste le docteur Etta Pisano lors d’une simple visualisation à l’aide d’un système de réalité augmentée développé à l’UNC, CRAIG et collab. [2009]

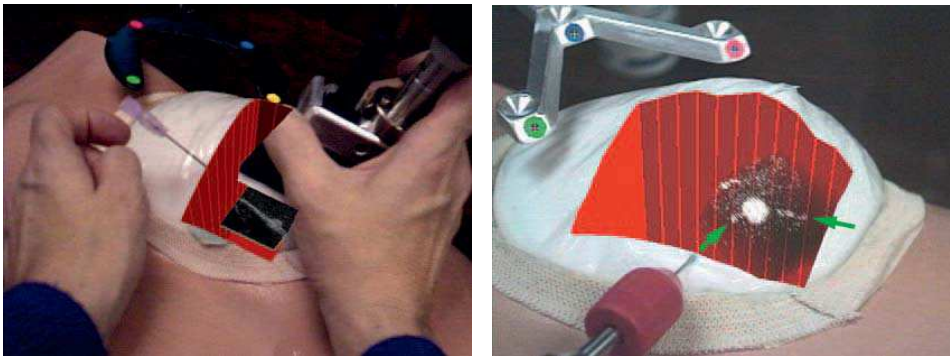


FIGURE 1.6 – Représentation virtuelle de l’intérieur du sein, CRAIG et collab. [2009]

1.3.2 Application à l’industrie

Un intérêt majeur de la réalité virtuelle : pouvoir se dispenser du réel en rendant perceptible une entité ou un événement à travers sa représentation virtuelle. Cet avantage fait de la réalité virtuelle un outil de conception assistée par ordinateur JIMENO et PUERTA [2007]. Il est souvent exploité dans l’industrie où une représentation virtuelle d’un produit peut servir à valider des choix de conception. A titre d’exemple, la société Caterpillar, fabricant des machines dans les domaines de construction des mines et de la forêt, a mis au point un système de prototypage virtuel VPS (figure 1.7) pour permettre à ses ingénieurs de valider des choix de conception avant de se lancer dans la production. Passer par une représentation virtuelle au lieu d’une maquette physique permet une économie sur les coûts de production mais également un gain de temps dans la boucle conception-validation d’un produit MUJBER et collab. [2004].

Le système de prototypage virtuel VPS CRAIG et collab. [2009] est constitué d’un système de visualisation stéréoscopique 3D immersif de type CAVE (Cave Automatic Virtual Environment). Le système comporte une plateforme physique avec un siège de machine, un volant, des pédales et d’autres éléments de contrôle pour manipuler le prototype du système d’une manière proche de la réalité. L’industrie de l’automobile utilise elle aussi la réalité virtuelle dans la production des voitures. ZIMMERMANN [2008] propose un état de l’art sur l’exploitation de la réalité virtuelle dans l’industrie d’automobile (figure 1.8).



FIGURE 1.7 – Système de Prototypage Virtuel de Caterpillar, CRAIG et collab. [2009]

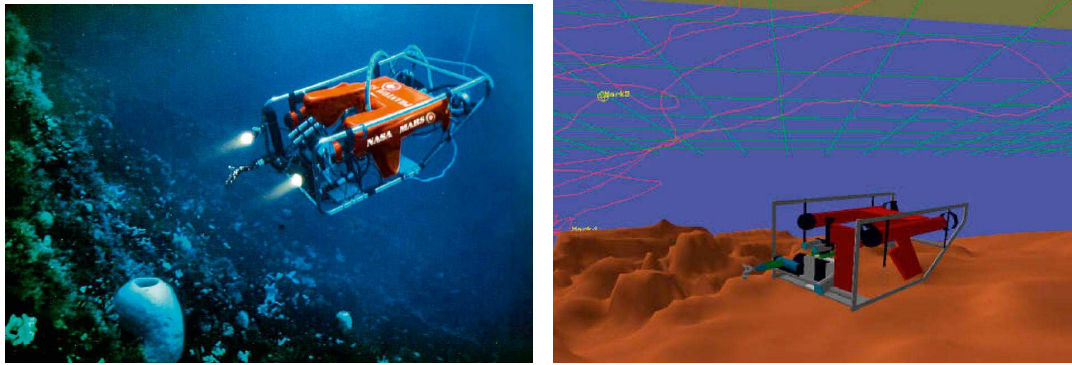


FIGURE 1.8 – Étude de l'habitacle d'une voiture dans un CAVE, ZIMMERMANN [2008]

1.3.3 Application à la science

La simple mention du nom "NASA" implique des images de l'aventure, des équipements de haute technologie et de la découverte des nouvelles frontières. Historiquement, la NASA a occupé les premiers rangs dans le développement des systèmes de réalité virtuelle. Le projet Intelligent Mechanism Group (IMG) en est un témoin. Ce projet de téléprésence a été utilisé pour aider les scientifiques à découvrir les surfaces planétaires et à contrôler les véhicules spatiaux. Les bénéfices sont une analyse plus générale, une meilleure compréhension spatiale et de nouveaux outils d'exploration (Visualisation sous plusieurs perspectives) et de manipulation. La mise en place du projet a été accompagnée par le développement d'un outil logiciel VEVI (Virtual Environment Vehicle Interface) permettant le contrôle d'une variété de véhicules dans différentes situations. Le premier véhicule contrôlé par cet outil, appelé "TROV" (Telepresence Remotely Operated Vehicle), a été exploité pour découvrir les profondeurs de l'océan Antarctique et recueillir des données sur les écosystèmes (figure 1.9)

Comme nous pouvons le remarquer dans la présente section, de nombreuses applications de réalité virtuelle ont été développées dans divers domaines. Ces applications bien qu'elles utilisent toutes la même technologie qui est la réalité virtuelle, sont multiples et très différentes. Chaque domaine se distingue par son propre type de données, ses besoins en interaction et ses utilisateurs spécifiques auxquels il s'adresse. Ces différences impliquent l'utilisation de plusieurs types d'interfaces ou encore des dispositifs d'entrée



(a) Les utilisateurs contrôlent à distance le robot via une téléprésence (b) Représentation virtuelle du robot et du fond de l'océan

FIGURE 1.9 – Un robot en train d'explorer les profondeurs de l'océan, CRAIG et collab. [2009].

et de sortie. Mais avant de s'intéresser aux interfaces rencontrées en réalité virtuelle, nous décrivons brièvement l'architecture d'un système dit de réalité virtuelle. Nous nous proposons d'en préciser les contours en présentant les différentes modules qui le compose.

1.4 ARCHITECTURE D'UN SYSTÈME DE RÉALITÉ VIRTUELLE

Le développement des applications de la réalité virtuelle nécessite un cadre de travail adéquat. Ce cadre devrait être extensible pour intégrer facilement de nouveaux appareils et de technologies. Malgré la variété de configurations matérielles et logicielles des environnements virtuels, ceux-ci partagent tous une architecture générale commune. Cette architecture est décomposée en plusieurs modules tels qu'ils sont présentés par la figure 1.10. Le module central d'un système de réalité virtuelle est ce que nous appelons le "moteur de réalité virtuelle".

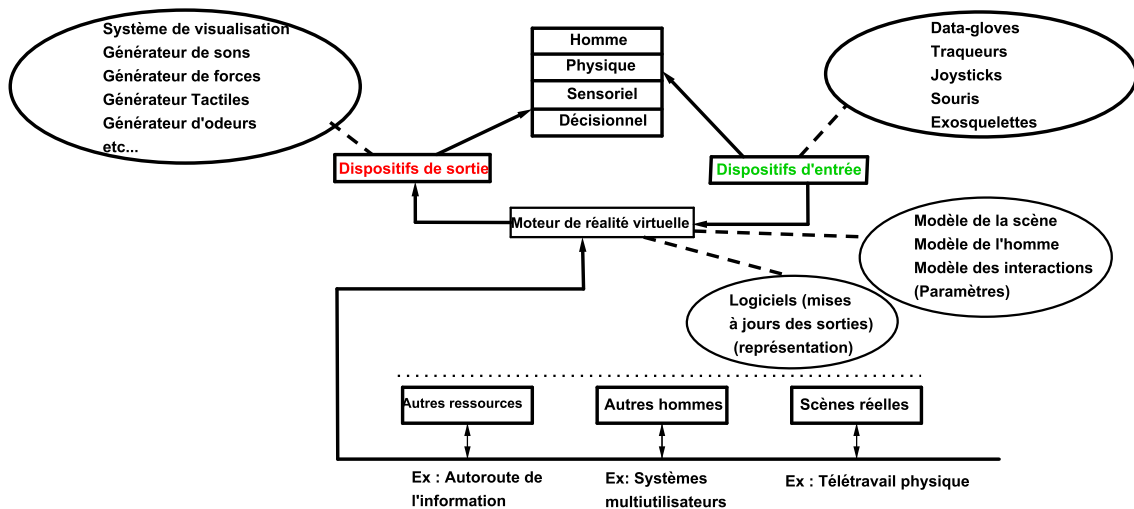


FIGURE 1.10 – Architecture d'un système de réalité virtuelle, COIFFET [1996]

Il s'agit d'un certain nombre de logiciels assurant deux fonctions principales :

- ✧ Une partie de ces logiciels sert à construire le mode virtuel désiré.

- ✧ D'autres logiciels garantissent la mise à jour convenable du monde virtuel en fonction des demandes de l'utilisateur.

Ce moteur est interfacé avec l'utilisateur qui agit dans le monde virtuel à l'aide des dispositifs d'entrée et qui reçoit les informations sensorielles émises par le monde virtuel par l'intermédiaire des dispositifs de sorties. Les dispositifs d'entrée et de sortie constituent alors des éléments majeurs de tout système de réalité virtuelle. Les dispositifs d'entrée déterminent la façon dont un utilisateur communique avec le système et de lui passer des informations dont il faut extraire les commandes. Ces informations peuvent être numériques (position et rotation, etc.), visuelles (par exemple, l'image d'une cible codée) et/ou auditives (signal audio). Quant à eux, les dispositifs de sortie permettent la présentation de l'environnement virtuel et des résultats de l'interaction à l'utilisateur. Nous pouvons considérer que les dispositifs de sortie sont les principaux responsables du sentiment d'immersion des utilisateurs dans l'environnement virtuel.

Nous allons maintenant rentrer plus en profondeur en présentant les différentes interfaces matérielles d'interaction utilisées en réalité virtuelle.

1.5 LES INTERFACES MATÉRIELLES DE RÉALITÉ VIRTUELLE

Le propos de cette section n'est pas d'effectuer l'inventaire de différentes interfaces matérielles rencontrées en réalité virtuelle mais plutôt de présenter les multiples choix qui s'offrent aux concepteurs des systèmes de réalité virtuelle. Ces interfaces constituent un élément crucial dans le développement des environnements virtuels. Elles sont utilisées comme des dispositifs d'entrée et/ou des dispositifs de sortie.

1.5.1 Les dispositifs d'entrée

Les dispositifs d'entrée regroupent essentiellement les deux familles suivantes d'interfaces : les systèmes de tracking et les interfaces gestuelles.

1.5.1.1 *Les systèmes de tracking*

Les systèmes de tracking ou "trackers" sont des dispositifs qui permettent d'effectuer le suivi d'un ou de plusieurs éléments dans un espace soit de 2D, soit 2.5 D (une position 2D dans un plan et une rotation), ou également de 3D (une position 3D dans l'espace) ou 6D (une position 3D et une orientation 3D dans l'espace). A titre d'exemple, ces éléments peuvent correspondre à la main et/ou à la tête de l'utilisateur. Ces systèmes sont basés sur différentes technologies ayant chacune des avantages et des inconvénients. **WELCH et FOXLIN [2002]** présentent en 2002 un état de l'art des différentes technologies de tracking. Bien que des améliorations technologiques soient apparues depuis, les auteurs décrivent les forces et faiblesses des différentes technologies et fournissent une base solide au choix de technologie pour le tracking. **MIHELJ et collab. [2014]** ont, quant à eux, proposé récemment un nouvel état de l'art sur les différents principes et technologies de tracking. Mais avant de se lancer dans la présentation de ces différentes technologies, nous précisons d'abord les performances d'un système de tracking.

1.5.1.1.1) Performances d'un système de tracking

Tous les systèmes de tracking, indépendamment de la technologie qu'ils utilisent, partagent des paramètres de performances comme la précision, la résolution, la répétabilité, la gigue, la dérive et la latence [BURDEA et COIFFET, 2003].

- ◆ La précision est la différence entre la position réelle de l'objet et la position mesurée par le système de tracking.
- ◆ La résolution est le changement minimum que le système de tracking peut détecter.
- ◆ La répétabilité : nous avons la même sortie pour la même entrée. Ainsi, les mêmes mouvements du système de tracking donnent les mêmes mesures.

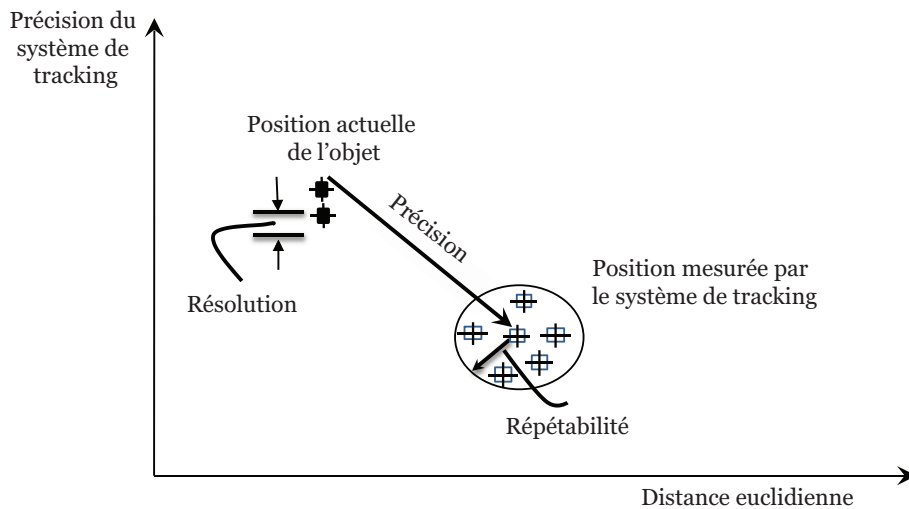


FIGURE 1.11 – Précision, résolution et répétabilité d'un système de tracking, [BURDEA et COIFFET, 2003]

- ◆ Plage de mesure est la plage où le système peut être exploité avec une précision acceptable. Elle s'exprime sous la forme d'une distance par rapport à l'origine de la référence du système de coordonnées.
- ◆ La gigue est la variation de la sortie du système de tracking au cours du temps quand l'objet à suivre est stationnaire. Elle peut être considérée comme le bruit du système de tracking.

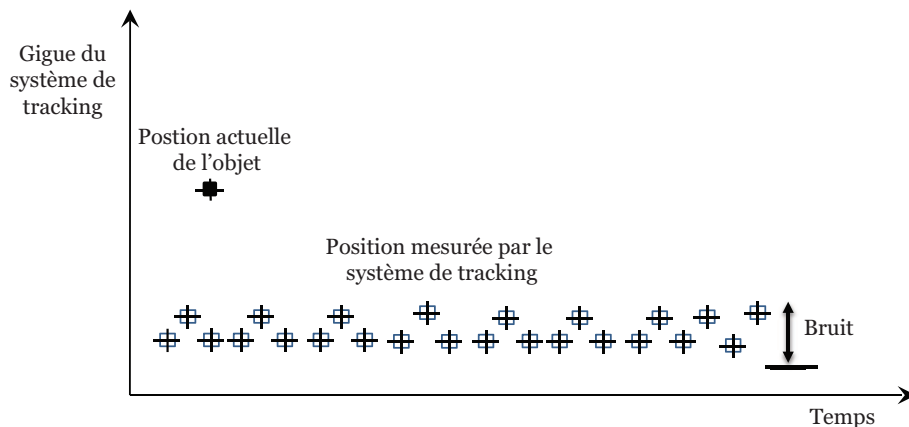


FIGURE 1.12 – Gigue d'un système de tracking, [BURDEA et COIFFET, 2003]

Un système de tracking sans gigue donne une valeur constante si l'objet est stationnaire. La gigue a des effets indésirables sur les graphiques : tremblement, des objets virtuels agités.

- ◆ La latence est le temps de retard entre l'action et le résultat ou encore le temps qui sépare le changement de position ou d'orientation et le moment où le système de tracking détecte ce changement. Cette propriété a plusieurs effets négatifs sur la simulation : Elle endommage l'expérience si elle est supérieure à 50ms (aucune immersion), elle introduit des incommodités si elle dépasse 10ms (figure 1.13).

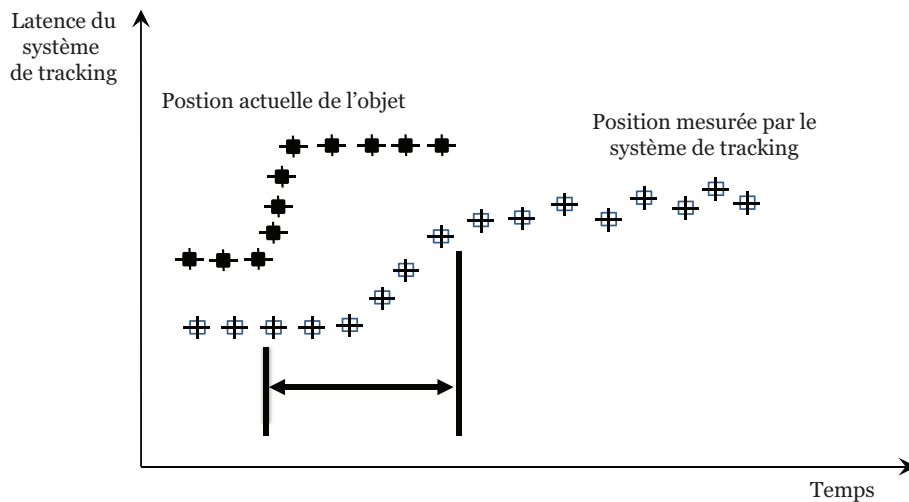


FIGURE 1.13 – Latence d'un système de tracking, [BURDEA et COIFFET, 2003]

- ◆ La dérive représente l'augmentation de l'erreur de tracking en fonction du temps. La précision du système de tracking diminue au fil du temps ce qui rend ses données inutiles. La dérive doit être contrôlée en la remettant périodiquement à zéro à l'aide d'un système de tracking secondaire qui ne souffre pas de dérive (figure 1.14).

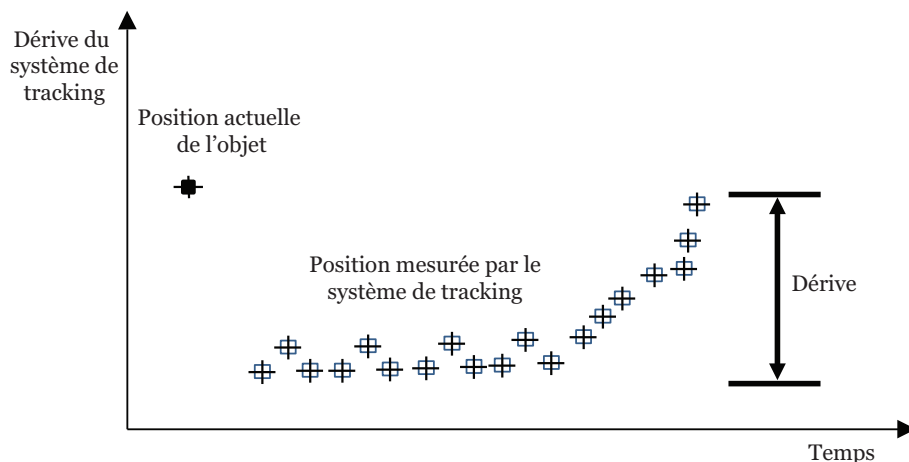


FIGURE 1.14 – Dérive d'un système de tracking, [BURDEA et COIFFET, 2003]

- ◆ Fréquence de mesure est le nombre de mesures retournées pendant une unité de temps. Plus la fréquence est grande plus la réponse de la simulation est meilleure. Typiquement nous prenons entre 30 et 240 échantillons par seconde. Cependant, si le système de tracking suit plusieurs objets, la fréquence d'échantillonnage peut baisser (figure 1.15).

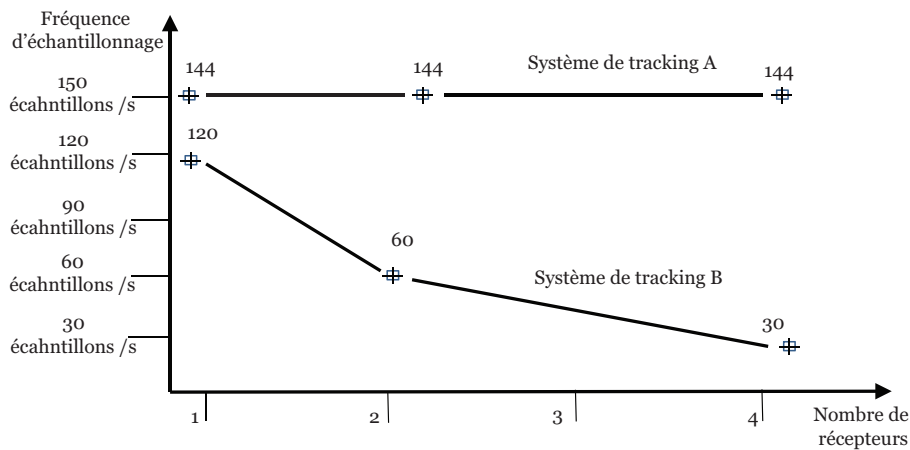


FIGURE 1.15 – La fréquence d'échantillonnage d'un système de tracking en fonction du nombre de récepteurs, [BURDEA et COIFFET, 2003]

En effet, le taux de transfert de données est limité et doit être partagé par plusieurs objets. À l'ajout de chaque nouvel objet, la bande passante par objet diminue.

Après avoir présenté les performances d'un système de tracking, nous allons maintenant décrire les différentes technologies de tracking utilisées dans la réalité virtuelle.

1.5.1.1.2) Les technologies de tracking

1.5.1.1.2.1 : Le tracking mécanique

Les systèmes de tracking mécaniques mesurent la position et l'orientation en utilisant une liaison mécanique directe entre le point de référence et l'objet à suivre. Généralement, ils sont composés de plusieurs segments articulés (figure 1.16). La position et l'orientation sont déterminées à partir des mouvements de rotation entre les différentes pièces mécaniques. Les principaux avantages de ce type de système sont la précision et la rapidité des mesures ainsi que leur très court temps de réponse. En revanche, leur inconvénient majeur est la contrainte de liberté de mouvement dans la mesure où le mouvement de l'utilisateur est limité à l'étendue du bras mécanique. Certaines versions des systèmes mécaniques peuvent calculer, en plus du mouvement, la vitesse et l'accélération généralement, en les associant à d'autres types de tracking.

1.5.1.1.2.2 : Le tracking magnétique

Quelque soit la technique de mesure de position et d'orientation utilisée, elle ne devrait pas être intrusive et en aucun cas limiter la liberté du mouvement de l'utilisateur. Compte tenu de ces revendications, les techniques de mesure sans contact ont largement remplacé les systèmes de tracking mécaniques. Les systèmes de tracking sans contact de nos jours utilisent le champ magnétique, l'ultrason, les caméras vidéo et infrarouges ou des accéléromètres et des gyroscopes. Nous commençons notre étude par les systèmes de tracking magnétiques puisqu'ils sont les plus répandus aujourd'hui. Les systèmes de tracking magnétique sont des dispositifs de suivi sans contact qui utilisent un champ magnétique produit par un émetteur stationnaire pour déterminer la position et l'orientation d'un élément récepteur mobile en temps réel. La plupart des systèmes ont les mêmes éléments de base. La figure 1.17 décrit les différents composants d'un système Aurora de Northern Digital (NDI). Les éléments clés d'un système de tracking magnétique sont Fi-

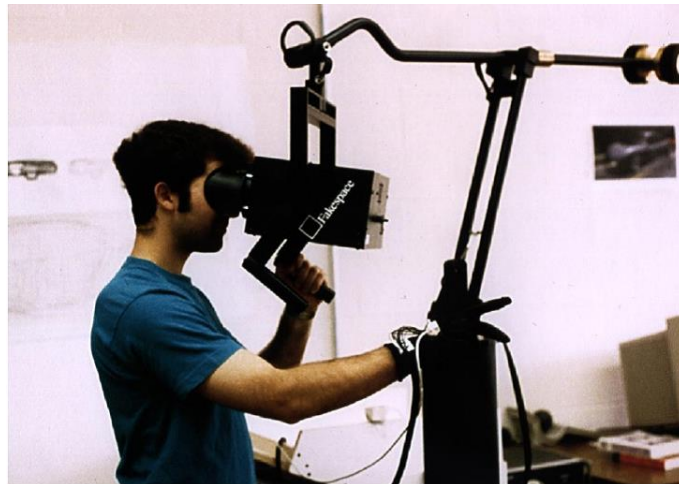
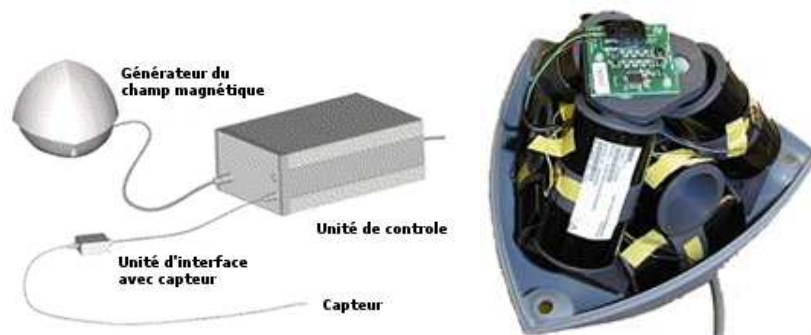


FIGURE 1.16 – Le BOOM (Binocular Omni-Orientation Monitor) , [FAKESPACE, 1994]

SCHER [2005] : un générateur de champ qui contient des bobines selon une configuration spécifique pour générer les champs magnétiques. Un capteur qui contient une ou plusieurs bobines qui produit des signaux électriques en présence des champs magnétiques induits. Une unité de contrôle qui prend en charge le contrôle le générateur du champ, qui reçoit les données en provenance du capteur et les synchronise avec la sortie de générateur du champ, effectue les calculs et assure la communication avec l'ordinateur. Certains systèmes, comme l'Aurora, ont une unité d'interface avec le capteur qui réalise l'amplification et la numérisation du signal reçu avant d'atteindre l'unité principale de contrôle.



(a) Les éléments d'un système magnétique d'Aurora (b) Générateur du champ magnétique

FIGURE 1.17 – Système Aurora, **FISCHER [2005]**.

Ce type de système de tracking a l'avantage d'être insensible aux obstacles visuels mais il a l'inconvénient d'être sensible aux interférences magnétiques qui peuvent être provoquées par des matériaux ferromagnétiques.

1.5.1.1.2.3 : Le tracking acoustique

Les systèmes de tracking acoustiques ont pour principe la mesure des distances déterminées par des temps de propagation de l'onde ultrasonore dans l'air. En effet, la position est calculée en connaissant le déphasage de temps entre l'émission et la réception de l'onde ultrasonique envoyée par les dispositifs acoustiques. En ce qui concerne l'orientation, elle est déterminée lors de l'utilisation de plusieurs récepteurs (au moins trois). Ce

type de système présente l'avantage d'être peu coûteux et léger à porter par l'utilisateur (figure 1.18). Au contraire des systèmes de tracking magnétique, ces dispositifs ultrasonores ont la possibilité de fonctionner en environnement métallique.



FIGURE 1.18 – Le système de tracking acoustique IS-900, [INTERSENSE, 1999]

Cependant, l'inconvénient majeur de ces systèmes est qu'ils exigent une vue directe entre l'émetteur et le récepteur. Si un obstacle vient bloquer la vue directe entre émetteur d'ultrasons et récepteur alors le signal de tracking est perdu. Aussi la présence du bruit de fond dans la chambre ou d'autres sources d'ultrasons peut corrompre le signal.

1.5.1.1.2.4 : Le tracking optique

Les systèmes de tracking optiques sont basés sur le traitement d'images prises par une ou plusieurs caméras. Ces caméras peuvent alors filmer des images dans la lumière visible ou bien dans le domaine des infrarouges. Dans ce qui suit nous présentons les périphériques de tracking optiques les plus utilisés.

1.5.1.1.2.4.1 Les caméras vidéo Il est courant de faire du suivi d'objet avec de caméras vidéo. Avec une seule caméra, nous sommes capables d'assurer sans aucun problème du suivi 2D, voir aussi du suivi 3D en utilisant des miroirs ou encore une variation des dimensions de l'objet à suivre. Toutefois, pour garantir un suivi 3D performant et efficace, qui permet de récupérer les six degrés de liberté, il nous faut un minimum de deux caméras. En effet, chacune des caméras voit l'objet sous un angle différent et par géométrie épi-polaire entre les deux plans images nous déterminons la position et l'orientation de l'objet. Les propriétés utilisées pour le suivi peuvent être sa couleur, sa taille, sa forme ou encore son déplacement. L'un des inconvénients d'un tel système est que le suivi est sensible à la présence d'autres objets identiques à l'objet que l'on désire suivre ou aux conditions d'éclairage. Mais, l'inconvénient le plus évident est celui de l'occultation de la caméra ou de l'objet à suivre. Quant-aux avantages, le système n'est pas coûteux et peut permettre, dans un environnement maîtrisé, un suivi très précis et robuste.

1.5.1.1.2.4.2 Les caméras infrarouges Différents systèmes utilisent des caméras infrarouges pour assurer le suivi d'objets. Pour garantir un bon suivi 3D de l'objet, comme pour le suivi à partir de caméras vidéo, il faut un minimum de deux caméras. Un des périphériques de tracking infrarouge le plus connu est celui fourni par **ART ADVANCED REALTIME TRACKING** [2016], son fonctionnement est détaillé dans la figure 1.19. Les

avantages des caméras infrarouges par rapport aux caméras vidéos est qu'ils fournissent un suivi de très haute précision et robuste. D'ailleurs, le fait de filmer en infrarouge permet d'isoler nettement mieux les marqueurs. Nous n'avons presque pas besoin de traitements supplémentaires, pour isoler seulement l'objet à suivre. Néanmoins, le suivi en infrarouge souffre, évidemment, du même problème d'occultation. Ce problème peut être en partie soulevé en rajoutant des caméras supplémentaires dans le but de permettre de couvrir le maximum de zones d'ombre. Cependant, ces systèmes sont très coûteux, au fur et à mesure qu'on leur rajoute des caméras supplémentaires.

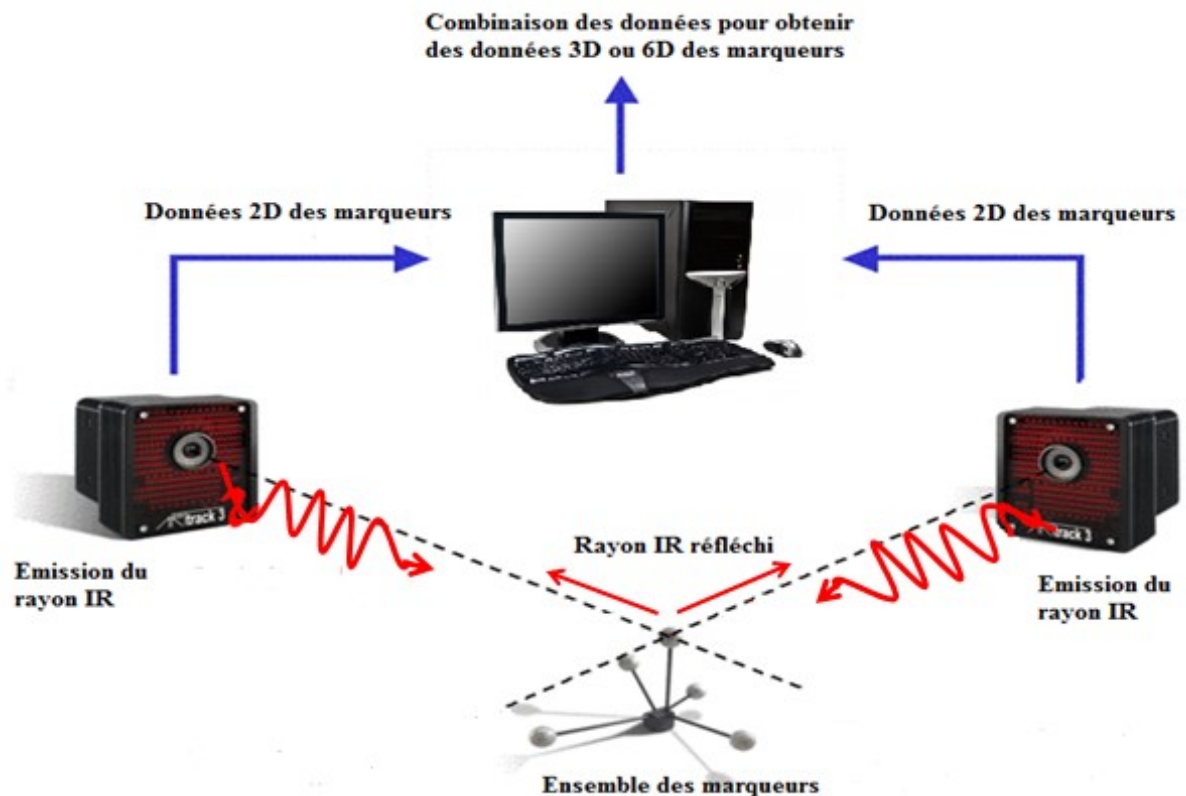


FIGURE 1.19 – Le principe fonctionnement du système optique, *ART ADVANCED REALTIME TRACKING* [2016]

1.5.1.1.2.4.3 Les centrales inertielles Ce type de système permet de mesurer à la fois l'accélération linéaire et la vitesse angulaire. Une centrale inertielle renferme trois accéléromètres et trois gyromètres. Un accéléromètre mesure la variation de la vitesse de déplacement suivant une direction et un gyromètre permet de mesurer la vitesse angulaire de rotation suivant un axe. En intégrant conjointement les vitesses angulaires des gyromètres autour de chaque axe, l'orientation des objets est obtenue. De la même façon, en intégrant les accéléromètres deux à deux, nous déduisons les variations de la position à une constante près à cause de la double intégration. L'avantage majeur de ces périphériques est qu'ils sont rapides, ne souffrent pas d'inférences électromagnétiques, ni du problème d'occultation et qu'ils sont peu encombrants (figure 1.20). Cependant, la position ne peut être déterminée que par double intégration et par conséquent une dérive dans l'estimation de la position. C'est pour cette raison que les périphériques inertiels sont toujours associés à d'autres capteurs de position plus lents dont ils servent à rectifier les mesures en raison de leurs temps de latence plus courts.



FIGURE 1.20 – La centrale inertielle MTx, XSENS [2016]

1.5.1.2 Les interfaces gestuelles

1.5.1.2.1) Les gants de données

La littérature foisonne des interfaces d'interaction qui sont basées sur les gants. La main étant la partie de notre corps que nous utilisons souvent pour interagir avec ce qui nous entoure. Les gants de données sont constitués de capteurs de flexions ou de l'écartement de doigts. L'ensemble de flexions récupéré nous indique la configuration de la main de l'utilisateur. Quelques gants comme les 5DT Data Glove Ultra (figure 1.21), développé par FIFTH DIMENSION TECHNOLOGIES [2016] sont dotés de capteurs optiques qui mesurent la quantité de lumière en sortie des fibres optiques montés sur chacun des doigts. La quantité de lumière observée reflète la flexion de chacun des doigts.



FIGURE 1.21 – Le Data Glove Ultra, FIFTH DIMENSION TECHNOLOGIES [2016]

D'autres gants, comme le CyberGlove (figure 1.22), de CYBER GLOVES SYSTEMS [2016] utilisent des matériaux sensibles à la flexion qui permettent de déterminer la configuration de la main.

D'autres gammes sont utilisées pour communiquer un retour d'information à l'utilisateur tel qu'un effort ou un retour vibro-tactile. En effet, le CyberGrasp (figure 1.23), de CYBER GLOVES SYSTEMS [2016] avec son exosquelette mis sur les doigts de l'utilisateur offrent la possibilité de ressentir la taille ou les contours d'un objet virtuel.

Afin de simuler les sensations de toucher de quelques matières, cette même société a conçu le gant CyberTouch (figure 1.24) de Cyber Gloves Systems CYBER GLOVES SYSTEMS [2016].



FIGURE 1.22 – Le CyberGlove, **CYBER GLOVES SYSTEMS** [2016]



FIGURE 1.23 – Le CyberGrasp, **CYBER GLOVES SYSTEMS** [2016]



FIGURE 1.24 – Le CyberTouch, **CYBER GLOVES SYSTEMS** [2016]

Ce dernier utilise des vibreurs qui lorsqu'ils fonctionnent dans une plage de fréquences précise, peuvent reproduire les micro-vibrations permettant la sensation de toucher certaines matières, comme l'eau, le bois ou l'acier.

Le majeur avantage de ces gants de données est qu'il est généralement aisé pour l'utilisateur de les exploiter pour autant qu'ils interagissent avec sa main comme il ferait dans la vie réelle. Néanmoins, la taille, l'encombrement ou le poids, causés par les câblages et les batteries font que ces dispositifs peuvent être rapidement considérés comme non ergonomiques ou très déplaisant à utiliser.

1.5.1.2.2) *La Wiimote*

La Wiimote est une manette de console de jeux-vidéo (figure 1.25), proposé par Nintendo[®] possédant différentes fonctionnalités. Elle est constituée de deux accéléromètres à trois axes, de multiples boutons, d'un haut-parleur, d'un vibreur et d'une connectivité bluetooth. Elle est équipée aussi d'une caméra infrarouge qui lui permet de se repérer dans l'espace lorsqu'elle est pointée en direction d'une source qui émette des infrarouges. Il existe une variété de bibliothèques de programmation permettant de faire dialoguer la Wiimote avec un ordinateur. Ceci a permis d'étendre l'utilisation de la Wiimote au domaine de la réalité virtuelle, offrant ainsi une interface peu coûteuse mais cependant très efficace.



FIGURE 1.25 – La Wiimote, NINTENDO [2010]

1.5.1.2.3) *La Kinect*

Kinect est un périphérique de console de jeu Xbox 360, développé par Microsoft dans le but de s'affranchir des manettes (figure 1.26). Elle repose sur l'usage d'une caméra 3D par stéréoscopie et triangulation active SMISEK et collab. [2011]. La Kinect se compose de :

- Quatre microphones qui sont combinés avec des convertisseurs analogiques numériques, utilisés pour la reconnaissance vocale et pour compenser l'acoustique des salles et atténuer le bruit. Le fonctionnement des microphones est le même que celui de l'oreille humaine : ils traduisent l'onde sonore émise par la source en un signal électrique. Quand l'onde sonore vient d'exciter le diaphragme du microphone, l'excitation de ce diaphragme va se traduire en signal analogique. La conversion du

signal analogique en un signal numérique est faite par un convertisseur analogique numérique ;

- Une caméra RGB (sensible aux couleurs : rouge, vert et bleu), standard avec un capteur photographique de type CMOS. Elle prend les images avec une fréquence de 30 Hz et en couleur 32 bits, avec une résolution VGA de 640x480 pixels ;
- Un émetteur-récepteur infrarouge pour calculer la profondeur d'un objet de résolution 320x240 pixels. Ce procédé est très rapide et offre une bonne résolution. Ainsi, il est possible d'acquérir un flux de données à 30 images per seconde. Les images capturées sont d'une taille de 640x480 pixels ;
- Un moteur qui permet de suivre la personne et d'orienter correctement l'espace du travail, en inclinant automatiquement la KINECT vers le haut pour voir la personne et vers le bas pour trouver le sol.

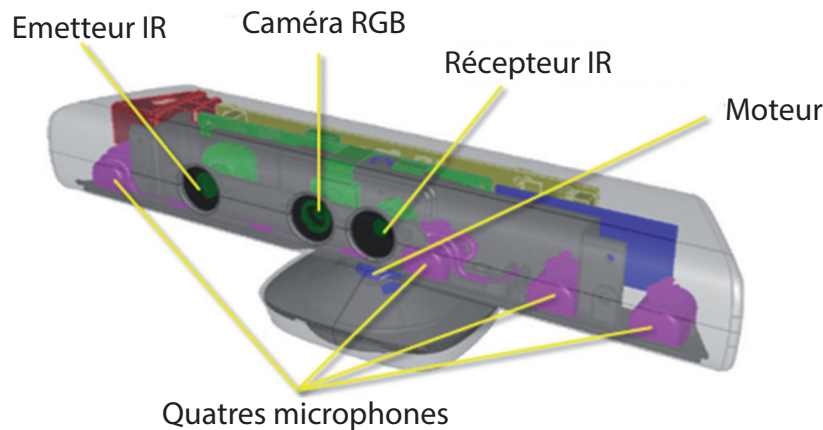


FIGURE 1.26 – Les éléments de la Kinect, MICROSOFT [2016]

Grâce à la connaissance du motif et de traitements propriétaires embarqués, la Kinect est capable de fournir une carte de profondeur dans le repère de la caméra infrarouge.

1.5.2 Les dispositifs de sortie

Les dispositifs de sortie permettent la présentation de l'environnement virtuel et les résultats d'interactions à l'utilisateur. Nous pouvons alors considérer que les dispositifs de sortie sont les principaux responsables de l'immersion de l'utilisateur dans l'environnement virtuel. Il s'agit principalement de présentations visuelles, auditives et haptiques.

1.5.2.1 Les interfaces visuelles

La majorité des dispositifs de sortie sont dédiés à l'affichage visuel de données. Un bon affichage peut être obtenu avec un large champ de vue, une très bonne résolution, et une haute fréquence de mise à jour. La bonne résolution permet d'obtenir des dispositifs de plus en plus larges offrant ainsi une image de meilleure qualité. La haute fréquence est essentielle pour éviter toute latence entre la mise à jour de l'environnement virtuel et l'affichage de l'information à l'utilisateur. En outre, un large champ de vue est primordiale afin d'assurer l'immersion par une large couverture du champ de vision. Dans cette section, nous présenterons quelques exemples d'interfaces visuelles.

Le CAVE **CRUZ-NEIRA et collab. [1992]** est reconnu comme étant la première interface immersive pour la réalité virtuelle (figure 1.27). Un utilisateur peut se déplacer à l'intérieur d'un cube à échelle humaine dont les six faces sont des écrans illuminés par rétro-projection. Sur ces écrans sont projetées deux images qui présentent deux points de vue légèrement décalés (de la distance interoculaire) d'une même scène. L'utilisation de lunettes adéquates donne la possibilité d'associer un point de vue à chaque œil pour garantir une vision stéréoscopique omnidirectionnelle à l'utilisateur. Par conséquent, ce type d'interface permet de parfaitement immerger visuellement l'utilisateur. Ce point pourra se révéler déterminant pour les simulateurs ou pour toute autre application qui requière un haut niveau d'immersion visuelle. Revers de la médaille, outre que ce type d'interface est complexe à configurer, il est très coûteux dû aux nombreux écrans, à la structure de l'interface mais également aux douze vidéoprojecteurs et aux multiples machines nécessaires pour contrôler le système. Par ailleurs, un espace d'installation présentant un volume important est requis.

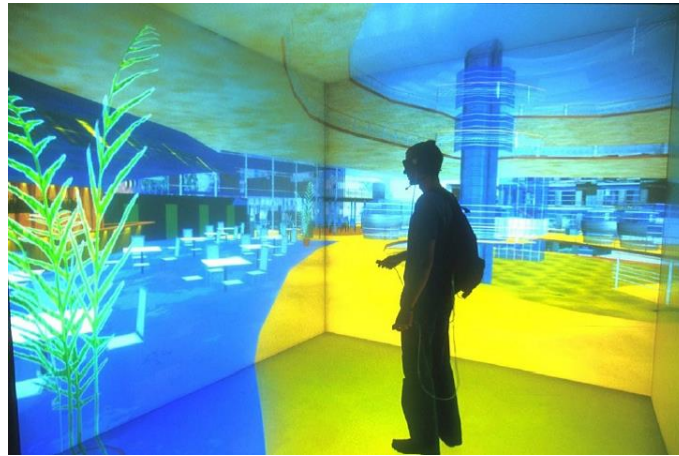


FIGURE 1.27 – Un CAVE à 5 écrans de projection, **IGATECHNOLOGIES [2016]**

Version améliorée du CAVE, le dénommé **C6 VIRTUAL REALITY APPLICATIONS CENTER OF IOWA STATE UNIVERSITY VRAC [2016]** développé au centre des applications de réalité virtuelle (Virtual Reality Applications Center) de l'Université de l'Iowa requière une surface de projection de cent millions de pixels (environ 16.7 millions de pixels par mur). Répartis sur les six faces d'un cube de trois mètres de côté, ces images revendiquent quarante huit stations de travail et vingt quatre projecteurs pour être affichées par rétro-projection (figure 1.28). Cette interface permet une vision stéréoscopique omnidirectionnelle de très haute résolution (4096x4096).

Le *SAS Cube* (figure 1.29) est une interface visuelle composée d'une surface de projection au sol de trois plans de projection verticaux. Ceux-ci peuvent être soit alignés pour offrir une grande surface de projection en ligne, soit en forme de U pour offrir une immersion plus enveloppante. Cette interface économise alors un plan de projection arrière et particulièrement la projection au plafond qui demande une salle au plafond élevé. L'idée est que la hauteur des plans est suffisamment grande pour recouvrir le champ de vision vertical de l'utilisateur et ainsi dissimuler la limite verticale de la zone de projection. Une interface de ce type a été inaugurée en octobre 2007 au **CENTRE DE LA RÉALITÉ VIRTUELLE DE LA MÉDITERRANÉE CRVM [2016]**. Cette dernière utilise des écrans de quatre mètres de haut pour trois mètres de large. Huit vidéos projecteurs garantissent la vision stéréoscopique.

Une solution alternative consiste à faire porter à l'utilisateur une visiocasque ou HMD



FIGURE 1.28 – Interface visuelle de réalité virtuelle, le C6. Des paires d'images sont rétro-projetées sur les faces du cube permettant ainsi une vision stéréoscopique omnidirectionnelle, **VIRTUAL REALITY APPLICATIONS CENTER OF IOWA STATE UNIVERSITY VRAC [2016]**

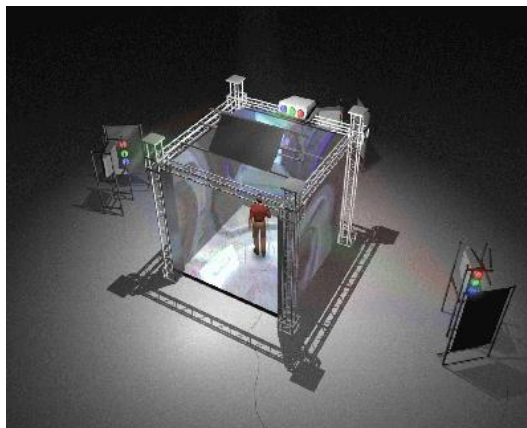


FIGURE 1.29 – Le SAS cube, **CLARTE [2016]**

(Head Mounted Device) (figure 1.30) qui suivant la position et l'orientation, lui permet donc d'orienter son regard dans toutes les directions. Les visiocasques ou HMDs, en général, prennent la forme de lunettes où les verres sont remplacés par deux écrans qui affichent deux images stéréoscopiques. Par exemple, le visiocasque nVisor de chez NVIS est relié à un boîtier qui récupère deux images non stéréoscopiques générées par la carte graphique de l'ordinateur (Mode affichage clone) puis ce boîtier génère à son tour deux images stéréoscopiques qui sont affichées sur les deux écrans du casque avec une résolution de 1280 -1024 pixels (figure 1.30).

Les visiocasques ont l'avantage de la simplicité d'installation par rapport aux CAVEs. Malgré leurs prix élevés, ils restent plus abordables que les CAVEs. Cependant, les visio-



FIGURE 1.30 – Un visiocasque de Nvis avec accusation optique de la position de la tête, Nvis [2016]

casques ont un inconvénient majeur qui est leurs poids qui peut se révéler très inconfortable pour une utilisation de longue durée. Un autre inconvénient est la contrainte d’être lié via un câble au boîtier ou à l’ordinateur.



FIGURE 1.31 – Les casques de Nvis, Nvis [2016]

1.5.2.2 Les interfaces à retour d’effort

Les interfaces haptiques regroupent les deux catégories d’interfaces suivantes : les interfaces tactiles et les interfaces à retour d’effort nommées aussi interfaces kinesthésiques. Les premières permettent de ressentir si une pression est exercée mais également la texture d’un objet, son volume, sa densité ou sa température. Les deuxièmes renvoient des informations sur les mouvements du corps mais aussi les forces exercées sur lui comme le poids par exemple.

Nous prenons le parti dans cette section de nous concentrer sur les interfaces à retour d’effort.

Ce type d’interface a été initialement utilisé pour la téléopération dans les milieux difficiles ou inaccessibles par l’être humain LÉCUYER [2001]. L’ambition était de donner à l’utilisateur les outils d’opérer directement au contact de l’environnement distant, faisant de cette interface une extension de son corps GOSSELIN [2005]. Puis avec le développement de technologies de l’informatique, les interfaces à retour d’effort sont exploitées dans les environnements virtuels.



FIGURE 1.32 – Système maître-esclave pour la téléopération, [GOSSELIN \[2005\]](#)

Différents critères ont abouti à différentes classifications des interfaces de retour d'effort. [HAYWARD et collab. \[2004\]](#) par exemple, a classifié ces interfaces selon leur nombre de degrés de liberté. Il distingue entre deux famille : Interface à peu de degré de liberté (pour 2 à 3 degrés) et interface à grand nombre de degré de liberté (pour 3 à 6 degrés).

[BERGAMASCO et AVIZZANO \[1997\]](#) ont proposé une autre classification en fonction de la nature de contact entre l'utilisateur et l'interface. Ils ont proposé cinq grandes familles : interface de type manette de console de jeux, interface à un seul point de contact, interface pour la réplication des tâches, interface à plusieurs points de contact et interfaces portables.

La classification que nous adoptons dans ce manuscrit est faite par rapport à l'architecture du système à retour d'effort [FUCHS et collab. \[2011\]](#). Nous distinguons deux grandes familles : fixes ou à réaction interne et portables ou à réaction externe. Concernant les interfaces fixes, les chaînes d'efforts se referment sur un socle fixe. Elles permettent de compenser le poids de l'interface et de contraindre les mouvements de l'utilisateur interagissant avec les objets virtuels. Néanmoins leurs espaces de travail sont limités. En revanche, pour les interfaces portables, la chaîne d'efforts se renferme sur l'utilisateur. Elles offrent donc plus de liberté de mouvement. Cependant leurs poids ne sont pas compensables et leurs mécanismes sont encombrants.

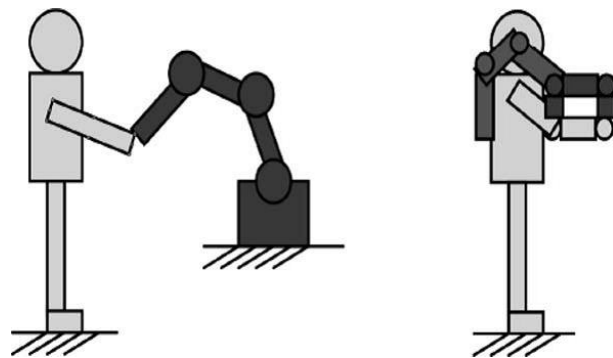


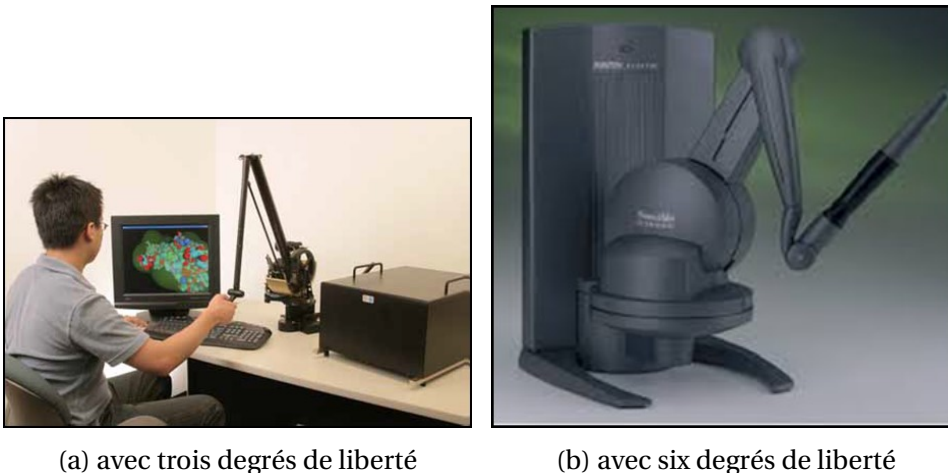
FIGURE 1.33 – Interfaces à retour d'effort à réaction interne et externe, [FUCHS et collab. \[2011\]](#)

1.5.2.2.1) Les interfaces fixes

Dans cette famille, les interfaces possèdent une base fixe ou indépendante de l'utilisateur. Le contact avec l'utilisateur s'effectue via un effecteur final. La structure de l'interface est de type sériel, parallèle ou spécifique. L'effecteur final renferme un préhenseur avec un seul point de contact ou un gant d'exosquelette pour l'utilisateur. Comme la structure de ces interfaces est fixée sur une base, l'utilisateur pourra ressentir une certaine rigidité sans être très gêné par le poids et l'inertie de l'interface.

1.5.2.2.1.1 : Structure série

Une solution performante et simple pour le retour d'effort est l'utilisation d'une structure série, telle que sur l'interface PHANTOM. Cette interface a été commercialisée depuis 1993 par l'entreprise Sensable Technologies et existe dans cinq modèles. Le PHANTOM Desktop, avec trois degrés de liberté actifs et trois degrés de liberté passifs, fournit un espace de travail de $16 \times 13 \times 13 \text{ cm}^3$ et le PHANTOM 6.0, possède 6 degrés de liberté actifs et un espace de travail de $19.5 \times 27 \times 37 \text{ cm}^3$ (figure 1.34).



(a) avec trois degrés de liberté

(b) avec six degrés de liberté

FIGURE 1.34 – PHANTOM Desktop, GEOMAGIC® HAPTIC DEVICES [2016].

Une interface de même famille est le Virtuoso développée par le CEA List et commercialisée dans plusieurs versions par la société Haption (figure 1.35). La version Virtuoso 3D15-25 associe un retour d'effort selon trois axes à une force maximale de l'ordre de 15N dans un volume de travail sphérique d'un diamètre de 250mm. La version Virtuoso 6D35-45, quant à elle, dispose de 6 degrés de liberté et peut fournir une force maximale de 35N dans un volume de travail sphérique dont le diamètre est de 450mm EMMANUEL [2005].

Les avantages de ce type d'interfaces sont la sensation de tenir un outil qu'ils procurent, ainsi que leur retour d'effort. Cependant, leurs espaces de travail sont souvent réduits. De plus, ils sont imposants et visibles, ce qui, généralement, diminue l'effet immersif lorsqu'ils sont utilisés devant un ou plusieurs écrans. Nous pouvons également noter leur manque de sensibilité au niveau du retour d'effort.

1.5.2.2.1.2 : Structure parallèle

Puisque les structures parallèles présentent des avantages importants, beaucoup d'interfaces à base fixe sont développées autour de cette architecture. Comme dans ce cas, l'inertie propre du système est limitée et que les actionneurs sont montés sur une base



(a) Virtuose 6D35-45



(b) Virtuose 3D15-25

FIGURE 1.35 – Virtuose, [HAPTION \[2016\]](#).

fixe, de telles interfaces sont plus simples, rigides et discrètes. En revanche, leurs espaces de travail sont plus petits proportionnellement à leurs dimensions. En outre, les points de singularité devront être éliminés de l'espace de travail.

SHaDe est une interface à structure parallèle possédant 3 degrés de liberté (figure 1.36). Elle est réalisée au Laboratoire de Robotique de l'université de Laval. Cette interface offre à l'utilisateur la possibilité de contrôler ses mouvements tout en ayant un retour d'effort. Elle se distingue par une conception simple et ergonomique et dispose des avantages comme celui de fournir une rotation autour d'un point situé dans la main de l'utilisateur, un espace de travail important et une manipulation précise avec le bras au repos [BIRGLEN et collab. \[2002\]](#).



FIGURE 1.36 – L'interface SHaDe, [BIRGLEN et collab. \[2002\]](#)

Le Cobot haptique est une interface pneumatique développé par [FAULRING et collab. \[2004\]](#) à l'université de Northwestern (figure 1.37). Cette interface est constituée de six actionneurs linéaires couplés sur un cylindre rotatif commun. Cette structure garantit une rigidité très élevée en minimisant les frottements et l'inertie.

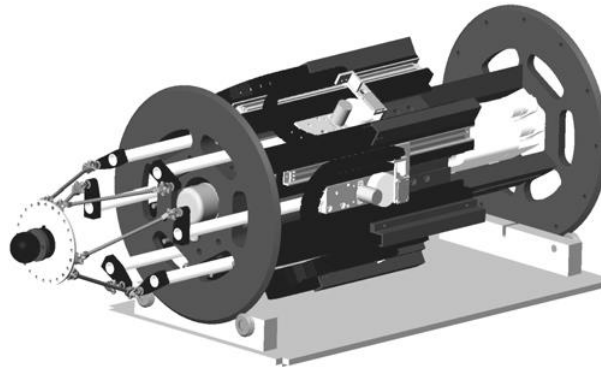


FIGURE 1.37 – Le cobot haptique, FAULRING et collab. [2004]

1.5.2.2.2) Les interfaces à câbles

Bien que les interfaces à câble soient des interfaces à structure parallèle, nous les avons rangés dans une nouvelle classe vue la multitude des recherches sur l'utilisation de câbles pour le retour d'effort. L'interface à câbles la plus célèbre est sans doute le SPIDAR (SPace Interface Device for Artificial Reality) conçue au Tokyo Institute of Technology. SATO [2002] l'ont présenté dans plusieurs versions : SPIDAR-I, SPIDAR-II, Networked SPIDAR, Scalable SPIDAR, SPIDAR-G, SPIDAR-8, SPIDAR-H . Toutes ces versions sont composées d'un certain nombre de blocs moteurs placés aux sommets d'une structure cubique et attachés au préhenseur à l'aide de câbles tendus qui s'enroulent sur des poulies entraînées par les moteurs. Puisque les câbles ne permettent que de tirer sur le préhenseur, il est indispensable d'utiliser au minimum un câble de plus que le nombre de degrés de liberté. La première version, SPIDAR I, produit un retour d'effort à trois dimensions pour un point de contact, soit un doigt d'une main en utilisant quatre câbles HIRATA et SATO [1992]. La deuxième version, SPIDAR II, emploie tous les sommets d'un cube pour classer huit blocs moteurs et produire deux retours d'effort à trois dimensions pour deux points de contact. Ceux-ci pourraient être soit deux doigts d'une main soit un doigt de chaque main ISHII et SATO [1993]. Les propriétés des interfaces SPIDAR I et SPIDAR II sont décrits dans le tableau 1.1.

TABLEAU 1.1 – Caractéristiques des interfaces SPIDAR I et SPIDAR II

	SPIDAR I	SPIDAR II
Espace de travail	900x600x600mm ³	300x300x300mm ³
Résolution en position	0.025mm	0.5mm
Effort crête	16N	4N
Effort continu	4N	-
Résolution en force	-	0.016N
Raideur apparente	1000 N/m	-

Le Networked SPIDAR a été conçue pour une interaction collaborative entre deux utilisateurs distants ISHII et collab. [1994]. Scalable-SPIDAR est associé à un grand écran pour les interactions à échelle humaine BUOGUILA et collab. [2000]. Avec huit câbles, SPIDAR-G propose sept degrés de liberté : trois translations, trois rotations et une préhension. Le préhenseur de cette version dispose de deux configurations pour fixer les câbles KIM et collab. [2002]. Plus récemment, ils ont conçu la version SPIDAR-8 qui comporte 24 blocs moteurs pour exercer le retour d'effort sur 4 doigts de chaque main (trois blocs

moteurs par doigt). SPIDAR-8 est employé dans les applications où la manipulation de chaque doigt doit être prise en compte. Cependant son espace de travail est très restreint faute du nombre de câbles (figure 1.38) WALAIRACHT et collab. [2002]. La dernière version de cette catégorie, SPIDAR-H, est conçue pour le CAVE (figure 1.39) et offre trois degrés de liberté pour un doigt de chaque main HASEGAWA et collab. [2006].

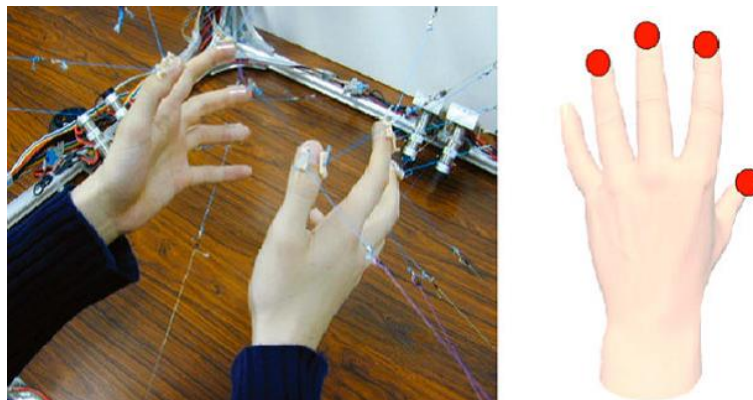


FIGURE 1.38 – L'interface à retour d'effort SPIDAR-8, WALAIRACHT et collab. [2002]



FIGURE 1.39 – L'interface à retour d'effort SPIDAR-H, HASEGAWA et collab. [2006]

Le principal avantage des interfaces du type SPIDAR est que nous pouvons ajouter des degrés de liberté en rajoutant des blocs moteurs aussi bien qu'agrandir leur espace de travail en déplaçant ceux-ci. L'utilisation de plus d'un certain nombre de blocs moteurs entraîne le croisement de câbles entre eux et des contacts supplémentaires avec l'utilisateur. Par ailleurs, la réponse et la précision du système se dégradent en utilisant des câbles assez longs.

1.5.2.2.3) *Intégration des interfaces à retour d'effort au sein des environnements à grande échelle*

Récemment, différentes applications industrielles ont été développées pour les environnements à échelle humaine, ce qui a motivé les chercheurs à intégrer le retour d'effort dans ces environnements. Plusieurs classes d'interfaces à retour d'effort ont été étudiées jusqu'ici. Les interfaces portables semblent plus cohérentes à utiliser dans de grands

espaces. Néanmoins, leurs poids doit être supporté par l'utilisateur. Pour résoudre ce problème, leur capacité en force est limitée. D'un autre côté, les interfaces liées à une base fixe ne sont pas adoptées à l'utilisation dans un espace à échelle humaine. En revanche, certaines versions de SPIDAR, particulièrement le Scalable-SPIDAR, ont été développées avec un espace de travail à grande échelle.

KUNZLER et RUNDE [2005] ont dressé une comparaison sur l'intégration des interfaces à retour d'effort dans les espaces à grande échelle. La figure 1.40 présente la synthèse de leur travail où les critères considérés sont les suivants :

- ✓ La taille de l'espace de travail est idéalement égale au champ de travail disponible à l'utilisateur. Elle peut lui être également plus élevée. Les caractéristiques du retour d'effort sont uniformes en tout point de cet espace ;
- ✓ La mobilité de l'espace de travail qualifie la dépendance de cet espace à la position de l'utilisateur. Dans le cas où la mobilité de l'utilisateur impose le déplacement du retour d'effort, l'utilisateur peut percevoir le décalage temporel du retour d'effort et la fidélité de l'interface se détériore ;
- ✓ La capacité multi utilisateur désigne la possibilité de collaboration simultanée de multiples utilisateurs avec des interfaces indépendantes ou une interface avec plusieurs effecteurs ;
- ✓ La raideur exposée par l'interface est essentielle pour simuler les surfaces rigides. Les interfaces portables sont incapables de retourner des valeurs de raideur élevées ;
- ✓ L'obstruction visuelle peut perturber l'immersion visuelle de l'utilisateur et par conséquent doit être réduite le plus possible ;
- ✓ La simplicité de l'opération est un critère qui comprend l'état du câblage, le poids de l'interface, le temps de réglage ainsi que le bruit engendré ;
- ✓ La fidélité du retour d'effort fournit une mesure qualitative du retour d'effort senti par l'utilisateur.

		Large Scale VE Suitability				Technical Criteria					Usability				LS Haptic Immersion
		Workspace Size	Workspace Mobility	Multi-User Capability	Evaluation Result	Displayed Stiffness	Force Level	Feedback		Evaluation Result	Visual Obstruction	Operation Ease	Haptic Fidelity	Evaluation Result	Evaluation Result
					Force (DOF)			Torque (DOF)							
passive-based	Linkage Type	①	○	○	○	●	●	● (3)	● (3)	●	○	●	●	①	①
	String Type (grounded)	●	○	① (2)	①	①	●	● (3)	● (3)	●	①	●	●	●	●
	Magnetic Levitation Type	○	○	○	○	①	●	● (3)	● (3)	●	○	●	●	①	①
	Mobile Platform Type	①	①	①	①	①	●	● (3)	● (3)	●	○	●	①	①	①
body-based	Exoskeleton Type	●	●	●	●	①	①	● (3+)	● (3+)	①	●	○	●	①	①
	String Type (portable)	①	●	●	●	①	●	① (1-3)	○ (0)	①	①	①	①	①	①
un-based	Jet Type	●	●	①	●	○	①	● (3)	● (3)	①	●	○	○	○	①
	Gyro Moment Type	●	●	●	●	○	○	○ (0)	● (3)	○	●	○	○	○	○

FIGURE 1.40 – Comparaison sur l'intégration des interfaces à retour d'effort dans les espaces à grandes dimensions, KUNZLER et RUNDE [2005]

1.5.2.3 Les interfaces auditives

La simulation de sources sonores est un paramètre complémentaire à la simulation visuelle et la simulation haptique pour la réalisation d'environnements virtuels, la perception sonore s'ajoutant à la perception visuelle et haptique pour générer un environnement plus naturel. Néanmoins, les travaux portant sur la simulation de sources sonores

virtuelles reste relativement peu important par rapport à ceux consacrés aux simulations visuelles et haptiques. Les interfaces auditives sont regroupées en deux grandes familles :

Les haut-parleurs souvent fixes, ils sont disposés dans la pièce. Ils permettent le partage d'information sonore par plusieurs personnes.

Les casques audio mobiles, ils sont portés par l'utilisateur. Ils permettent l'individualisation des informations sonores par chaque personne.

Les rendus sonores peuvent être :

- ◆ **Mono-auraux** : un seul haut-parleur est utilisé. Ceci est suffisant, pour produire des sons simples comme des alarmes par exemple.
- ◆ **Stéréophoniques** : nous utilisons un casque ou encore deux haut-parleurs. Le rendu est partiellement spatialisé (droite-gauche).
- ◆ **Spatialisés** : nous utilisons deux haut-parleurs ou plus pour placer une source sonore virtuelle n'importe où dans l'espace. Un casque audio peut être aussi exploité, mais si l'utilisateur bouge la tête, il faut, en temps réel, récupérer la position de la source ce qui oblige à utiliser un localisateur.

CONCLUSION

Dans cette première partie, nous avons remarqué la difficulté à donner une définition unique à la réalité virtuelle. En effet, la réalité virtuelle s'appuie sur de multiples domaines de recherche. Ceci a pour conséquence que la réalité virtuelle a du mal encore à exister en tant que discipline à part entière. Suite à cet état de l'art sur les définitions de la réalité virtuelle, nous nous sommes intéressés aux différents domaines d'applications. Nous avons porté ensuite notre attention sur les interfaces matérielles rencontrées en réalité virtuelle. Toutes ces interfaces sont développées autour des différentes technologies ayant chacune leurs avantages et inconvénients. Ceci nous a permis d'aborder les principaux problèmes inhérents à ces interfaces sous des angles différents. Néanmoins, le problème commun à toutes ces interfaces est la calibration. En effet, afin de parvenir à une exploitation plus efficace, il est impératif de connaître leurs propriétés, avantages et inconvénients. Le but est de proposer, dans la mesure de possible, un moyen d'améliorer la confiance accordée à ces interfaces ainsi que leurs performances.

Chapitre 2

Calibration d'un système de tracking en réalité virtuelle

Résumé

Ce deuxième chapitre est consacré à l'exploration des méthodes utilisées pour la calibration des systèmes de tracking de réalité virtuelle. La calibration est cruciale pour garantir une bonne qualité de tracking. En effet, elle doit être effectuée avec soin puisque son issue peut affecter les performances et la précision des mesures retournées par le système. La première section propose de revenir sur les protocoles expérimentaux permettant la caractérisation d'un système de tracking. Dans une deuxième section, nous présentons les différentes techniques proposées pour corriger les erreurs de tracking. Cet état de l'art ne nous limite pas uniquement au domaine de la réalité virtuelle, mais nous amène souvent à explorer d'autres domaines de recherche comme par exemple le domaine de la chirurgie guidée par image ainsi que le domaine de la robotique.

Sommaire

Introduction	39
2.1 Les protocoles expérimentaux de calibration des systèmes de tracking .	39
2.1.1 Supports physiques/virtuels pour l'acquisition des mesures de référence	40
2.1.2 Systèmes de tracking complémentaires pour l'acquisition des mesures de référence	44
2.2 Classification des protocoles de collecte de données	52
2.2.1 Collecte de données dynamique versus statique	52
2.2.2 Collecte de données automatique versus manuelle	52
2.3 Classification des erreurs	53
2.3.1 Les erreurs statiques	53
2.3.1.1 Distorsions statiques	53
2.3.1.2 Les erreurs aléatoires ou Gigue (Jitter)	53
2.3.2 Les erreurs dynamiques	54
2.3.3 L'erreur totale de tracking	55
2.4 Les techniques de calibration	55

2.4.1	L'interpolation tri-linéaire	55
2.4.2	L'interpolation par le point le plus proche	57
2.4.2.1	Méthode de pondération linéaire	57
2.4.2.2	Méthode de pondération exponentielle	57
2.4.2.3	Méthode de pondération basée sur le carré de la distance	58
2.4.2.4	Autres méthodes de pondération	58
2.4.3	Interpolation polynomiale	58
2.4.4	Les polynômes de Bernstein	60
2.4.5	Technique de Hardy HMQ (Hardy's multi-quadric method)	60
2.4.6	Technique basée sur les réseaux de neurones	61
2.5	Comparaison des techniques de calibration	63
	Conclusion	64

INTRODUCTION

Les systèmes de tracking sont devenus indispensables depuis qu'ils sont exploités majoritairement comme dispositifs d'entrée dans les systèmes d'interaction 3D. Le rôle d'un système de tracking est de retourner les coordonnées d'une entité dans l'espace 3D : soit uniquement la position, soit la position et l'orientation. Dans le cas d'un système de tracking parfait (comme nous l'avons mentionné dans la section 1.5.1.1.1), la position de l'entité retournée par le système est égale à la position réelle de cette dernière dans l'espace 3D. Néanmoins dans la réalité, il n'existe pas un système idéal et en conséquence, un décalage peut exister entre les coordonnées perçues par le système et les vraies coordonnées. Ces erreurs sont généralement fonction dépendante de la distance de l'entité à suivre par rapport aux sources de tracking. Ces erreurs peuvent souvent être corrigées. Nous appelons cette procédure : la calibration.

2.1 LES PROTOCOLES EXPÉRIMENTAUX DE CALIBRATION DES SYSTÈMES DE TRACKING

Pour caractériser la distribution des erreurs, nous avons besoin d'un protocole expérimental permettant l'acquisition des mesures. En effet, il existe une multitude de techniques qui permettent de recueillir les données d'un système de tracking. Certaines de ces techniques feront l'objet de la section suivante. Néanmoins, la technique optimale n'est pas nécessairement universelle mais plutôt elle est étroitement liée au type de l'application en question. Les méthodes de calibration se distinguent par le type d'équipement utilisé et par le modèle mathématique appliqué pour la caractérisation et la correction des erreurs qui entachent les mesures. Différents équipements ont été utilisés, nous citons les supports physiques ou virtuels ou encore les systèmes de mesure complémentaires tels que les systèmes de tracking optiques ultrasoniques ou mécaniques. Toutes ces méthodes utilisent deux séries de mesures :

$$\begin{aligned} \{p_i\} &\subset P \subset \mathbb{R}^3 \\ &\text{et} \\ \{q_i\} &\subset Q \subset \mathbb{R}^3, i = 1, \dots, n \end{aligned}$$

Où $\{p_i\}$ désigne l'ensemble de mesures de "tracking" dans l'espace de suivi P (entaché d'erreurs), $\{q_i\}$ désigne l'ensemble de mesures de référence dans l'espace vrai Q (non entaché d'erreurs). Certaines techniques de correction d'erreurs imposent souvent certaines restrictions sur la topologie des mesures p_i et q_i . Les mesures doivent être prises selon une mire soit dans l'espace de suivi ou l'espace vrai. Plusieurs approches ont été développées. Une première approche consiste à déplacer le système de "tracking" dans une mire à trois dimensions rectilinéaire dont les coordonnées des nœuds sont préalablement connues dans l'espace vrai et à enregistrer les coordonnées correspondantes retournées par le système de tracking. Dans la deuxième approche, le système de tracking est déplacé dans une mire à trois dimensions rectilinéaire dans l'espace de suivi et les coordonnées vraies correspondantes du système sont enregistrées. Une troisième approche consiste à déplacer le système de tracking d'une façon aléatoire ou irrégulière et à enregistrer à la fois les coordonnées vraies et mesurées par le système tracking à chaque emplacement puis les ré-échantillonner sous la forme d'une mire à trois dimensions rectilinéaire.

2.1.1 Supports physiques/virtuels pour l'acquisition des mesures de référence

Un support physique typique (figure 2.1), souvent utilisé pour l'acquisition des mesures des systèmes de tracking magnétique, est constitué d'une planche de bois ou de plexiglas perforée et montée sur une charpente de bois mobile. Les perforations permettent de fixer le capteur à un endroit dont les coordonnées, sont préalablement connues.

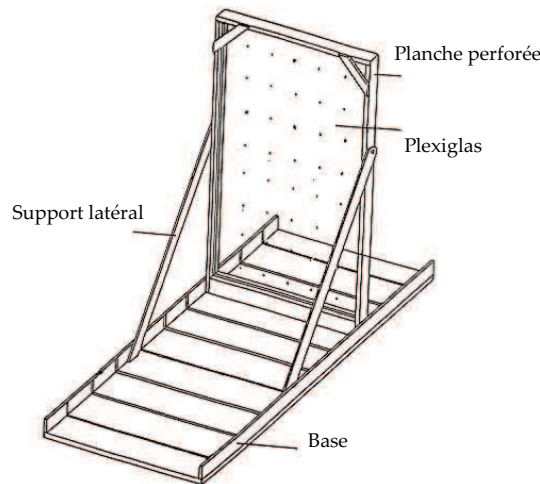


FIGURE 2.1 – Exemple d'un support physique typique

Dans les travaux de BRYSON [1992], un panneau perforé de 6x4 pieds est construit pour recueillir un ensemble de positions du système de tracking. Le plancher du laboratoire a été marqué avec des positions pour couvrir un volume régulier de 8x8x6 pieds. Certains trous sont marqués sur le panneau à des intervalles de douze pouces pour constituer des sites de mesure. La table de conversion (Look Up Table : LUT) est remplie en déplaçant le capteur à chaque site et en y prenant des mesures. Dans ce protocole d'acquisition, les mesures de références q_i correspondent aux coordonnées des trous perforés. Les mesures p_i correspondent aux valeurs retournées par le système de tracking à chacun des trous perforés. Plusieurs mesures (soixante mesures) ont été prises pour calculer la position moyenne p_i et l'écart type. Un ensemble de mesures de validation (des sites de mesures décalés de six pouces par rapport aux précédents) différent des mesures qui ont servi à la correction est utilisé pour évaluer les performances des différentes techniques de correction. Seules les erreurs inférieures à cinquante pouces ont été retenues, soit cent cinquante points sur les deux cents cinquante six possibles. Ce protocole est relativement précis mais il prend beaucoup de temps pour être appliqué car le capteur doit être déplacé manuellement deux cents cinquante six fois dans tout le volume. Par ailleurs, aucune modification de l'orientation n'est possible durant l'acquisition des données.

Une autre technique a été proposée par KINDRATENKO [1999]. Elle consiste à un tube de plastique télescopique de longueur variable (2, 3, 4, 5,6 et 7 pieds) pour fixer le capteur et une charpente en bois pour soutenir le tube. Le plancher du système de réalité virtuelle CAVE a été marqué par une série de lignes perpendiculaires espacées de 1 pied pour former une mire à deux dimensions rectilinéaire (figure 2.2). Cette mire est utilisée pour le repérage du déplacement de la charpente de bois et ainsi former un ensemble de mesures de références dans l'espace vrai. Le système de tracking est déplacé par incrément jusqu'à ce que tout le volume de CAVE soit couvert. La table de conversion est

construite en déplaçant le capteur à cent sites de mesures q_i puis en enregistrant les mesures p_i retournées par le système de tracking. Les mesures hors champ ont été éliminées de la table ce qui réduit la taille de la table à quatre vingt enregistrements. Un ensemble de soixante quatre valeurs a été utilisé pour la validation des techniques de correction. La précision des méthodes d'acquisition manuelle des mesures est estimée à ± 1 centimètre et ± 1 degré.

Bien que cette technique soit relativement précise, elle prend beaucoup de temps à être mise en œuvre car encore une fois le capteur doit être déplacé manuellement. L'orientation du capteur doit aussi demeurer fixe tout au long de l'expérience [KINDRATENKO \[2000\]](#).



FIGURE 2.2 – Equipements utilisés pour l'acquisition de la mire de calibration, [KINDRATENKO et SHERMAN \[2005\]](#).

Une année plus tard, ce même protocole a été utilisé pour l'acquisition et la correction des orientations [KINDRATENKO et BENNETT \[2000\]](#). La table de correspondance est remplie en déplaçant le capteur à pas régulier à des emplacements dont les coordonnées en position et en orientation sont déjà connues. Après un alignement soigneux, la précision de cette technique de mesure manuelle est de l'ordre de 0,5 centimètres et 1 degré. De meilleurs résultats pourraient être réalisés avec une table plus dense en termes de nombre de points, plus particulièrement pour ce qui est des mesures aux frontières du volume de mesure. La résolution d'un pied n'est pas suffisante pour avoir un bon modèle d'interpolation. Il serait indispensable d'augmenter la densité des points situés loin de la source de tracking magnétique.

Dans le protocole d'acquisition utilisé par [BRIGGS \[1999\]](#), les mesures sont recueillies par plusieurs capteurs électromagnétiques montées sur un montage fait de PVC et assemblé sur une base en bois. La technique de correction exige que les mesures p_i soient prises d'une manière rectiligne dans l'espace de suivi, cependant cette méthode d'acquisition produit un résultat contraire. Pour cela les données ont été ré-échantillonnées sous la forme d'une mire rectilinéaire dans l'espace de suivi. Chaque point de la mire p_i est entouré par d'autres points de mesure formant un ensemble de tétraèdres. Puisque les données sont disponibles dans les espaces de suivi P et l'espace vrai Q, les vecteurs d'er-

reur sont calculés. Le vecteur représentant l'erreur de position est utilisé à chaque point de mesure dans l'espace P pour interpoler et former le tétraèdre dans l'espace Q.

ZACHMANN [2000] a utilisé une colonne de bois fixée sur une charpente de bois. Plusieurs capteurs sont attachés à cette colonne verticalement à des endroits bien définis. En déplaçant la colonne aux marqueurs situés sur le plancher, les différents capteurs se trouvent à des positions connues dans l'espace vrai.

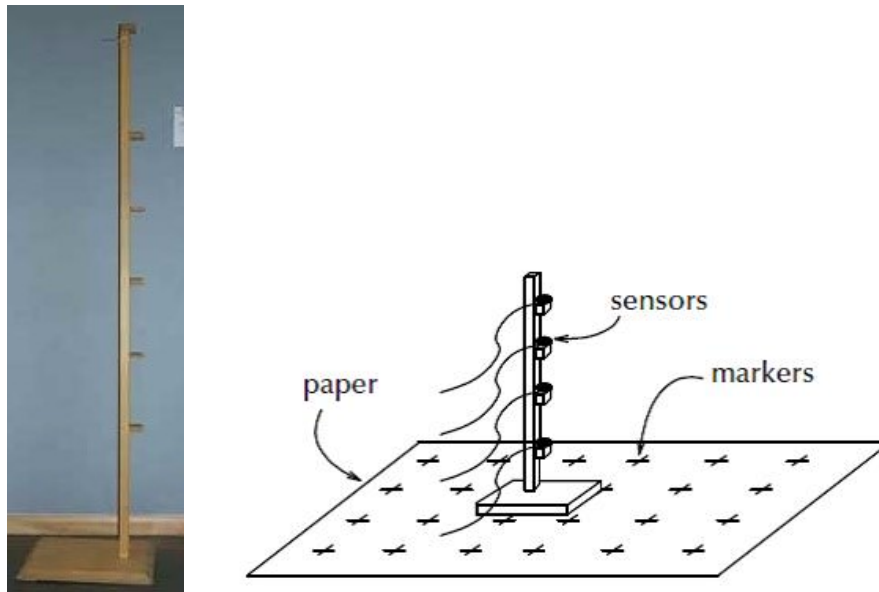


FIGURE 2.3 – Schéma de la plateforme expérimentale, ZACHMANN [2000]

Avec le montage décrit ci-dessus, 144 (6x6x4) mesures ont été enregistrées dans environ vingt minutes. En effet, quatre positions sont enregistrées au même temps. Cette technique d'acquisition est relativement précise et considérablement rapide par rapport aux autres techniques utilisées par Bryson et Kindratenko.

Dans les travaux DAY et collab. [2000], une technique d'acquisition d'un système de tracking magnétique de Fastrak a été présentée. Le protocole a été repris des travaux antérieurs des auteurs DAY et collab. [1998]. Les schémas de la figure 2.4 montrent les principaux éléments constitutifs du protocole. Une planche perforée et un gabarit de dimensions 0,8m x 1,2m x 1,4m ont été mis au point. Le protocole est conçu de manière à ce que le capteur puisse être positionné à trois cents soixante positions et selon douze orientations et qui permettent une résolution de vingt centimètres entre les sites de mesures.

Les travaux d'OCHOA-MAYORGA et collab. [2005] présentent une technique d'acquisition de mesures en orientation d'un système de mesure électromagnétique à plusieurs capteurs. L'environnement de mesure est une salle de chirurgie fortement bruitée par les distorsions électromagnétiques. Le système de tracking est un ensemble de douze capteurs Ultratrak de l'entreprise Polhemus. Deux séries d'enregistrements ont été prélevées pour chaque capteur, une première série pour la correction et une seconde pour la validation. Pour chacune des séries, quatre cents mesures ont été enregistrées à chaque position d'un volume de mesure rectiligne. Ensuite, la position et l'orientation moyenne du capteur sont exploitées. La position moyenne est la moyenne statistique des quatre cents mesures alors que l'orientation moyenne est transposée en quaternion. Au début, le volume de mesure est segmenté en cubes rectilignes de 0,20 m x 0,20 m x 0,10 m, (figure 2.5). La série d'enregistrement de validation a été découpée en divisions de tailles identiques, mais avec un léger décalage de 0,10 m x 0,10 m x 0,5 m.

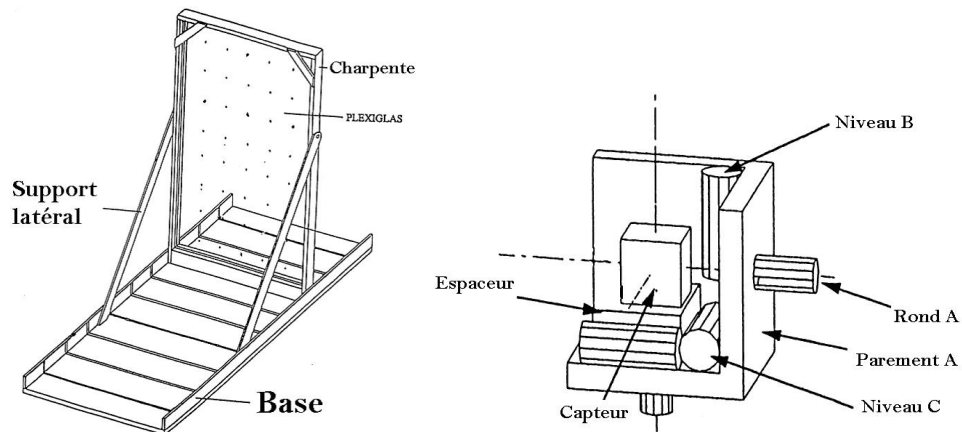


FIGURE 2.4 – Schéma des principaux éléments de montage expérimental, DAY et collab. [1998]

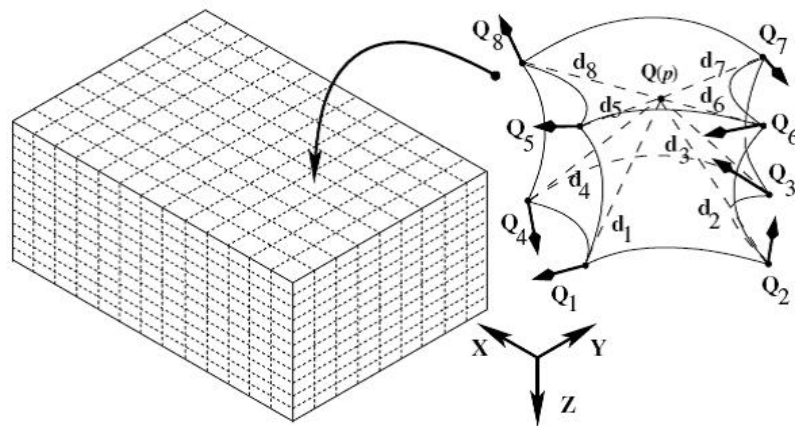


FIGURE 2.5 – Le volume de calibration, OCHOA-MAYORGA et collab. [2005].

HUMMEL et collab. [2005] et J HUMMEL AND M FIGL AND W BIRKPELLNER AND M R BAX AND R SHAHIDI AND C R MAURER JR AND H [2006] conçoivent une plateforme de mesure en polycarbonate de dimensions $0,55m \times 0,55m \times 0,0147m$ pour l'analyse des systèmes de tracking. Une mire de 10×10 trous a été percée soigneusement de telle sorte que deux sites de mesure adjacents soient séparés de cinquante millimètres. L'extrémité du capteur magnétique peut être fixée de manière rigide avec des vis non métalliques à trois positions. Au centre de la plateforme, un anneau de trous peut être utilisé pour les mesures d'orientations. Le capteur peut être donc monté avec un pas de progression de 11.25 degré entre les paires de trous adjacents de façon à mesurer les 360 degrés.

HIMBERG et collab. [2012] et HIMBERG et MOTAI [2014] proposent un système composé de deux dispositifs : un dispositif d'interpolation (Interpolation fixture) et un dispositif de cartographie (Mapping fixture). Le premier dispositif est un petit cube équipé de capteurs électromagnétiques sur chacune de ses sommets. Le deuxième dispositif est similaire à une baguette ou bâton construite à base de matériau non ferreux, sur lequel sont placés des capteurs à intervalle régulier. Les deux dispositifs sont utilisés conjointement pour recueillir un nuage de points.

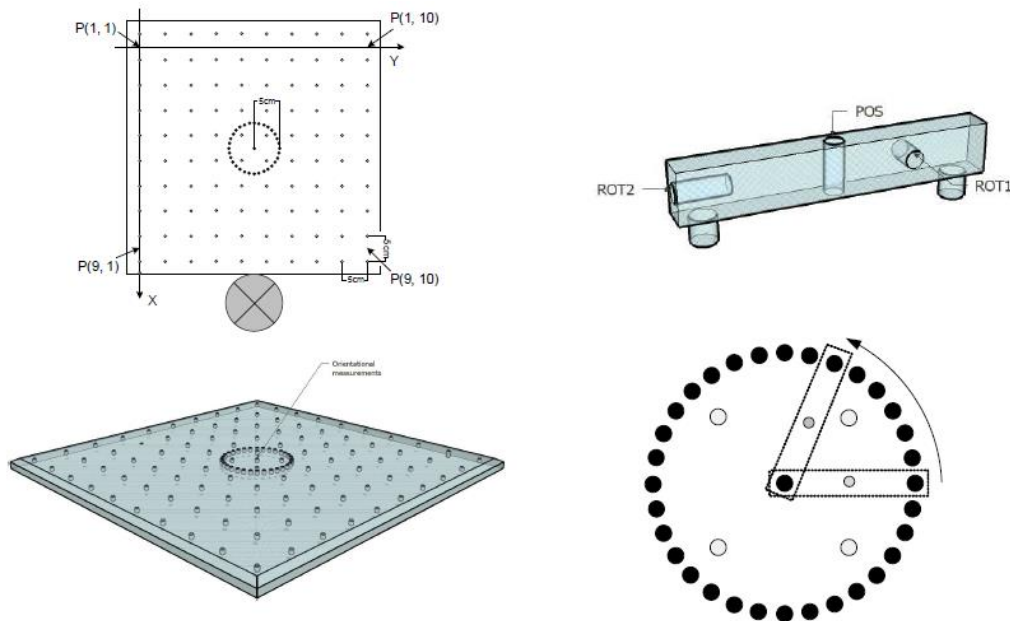


FIGURE 2.6 – La plateforme de mesure en polycarbonate introduite par [HUMMEL et collab. \[2005\]](#).

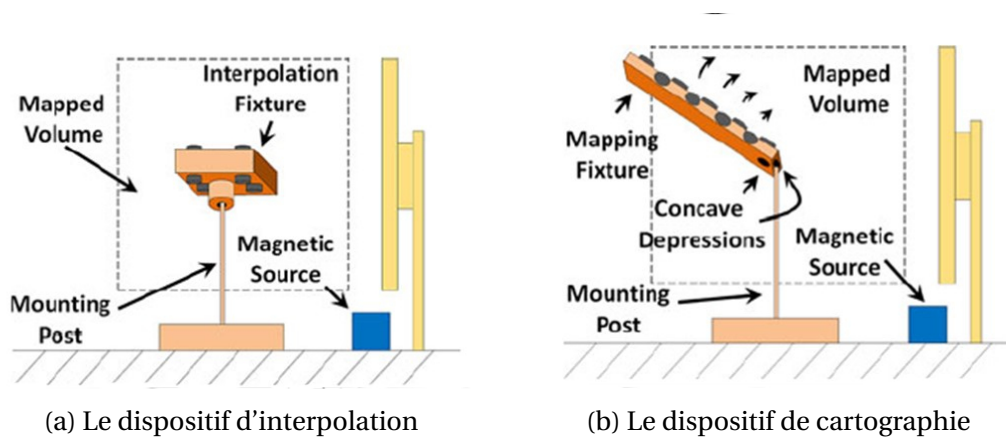


FIGURE 2.7 – Protocole volumétrique pour la calibration des systèmes de tracking électromagnétiques, [HIMBERG et collab. \[2012\]](#)

2.1.2 Systèmes de tracking complémentaires pour l'acquisition des mesures de référence

Dans les travaux [GHAZISAEDY et collab. \[1995\]](#), un système de tracking à ultrason (UMD : Ultrasonic Measuring Device) a été utilisé pour mesurer la position du capteur électromagnétique dans l'espace vrai. Parmi les équipements utilisés, une interface visuelle de réalité virtuelle CAVE, un système de tracking électromagnétique de type Flock of Birds, un système de tracking à ultrason de précision 1,5% sur une distance de un pied comprenant quatre transducteurs ultrasoniques, des lunettes stéréoscopiques et un ordinateur équipé d'un logiciel permettant l'obtention des positions mesurées par le système acoustique. Le volume de mesure, situé à l'intérieur du système de réalité virtuelle, est rempli de cubes de un pouce à intervalle régulier de un pied. Les points de mesures sont principalement concentrés au centre du volume. Les mesures de système de tracking ultrasonique dont l'angle d'inclinaison ou de roulis est supérieur à 10 degrés ont été exclues. Moins de quatre cents mesures ont été récupérées au total.

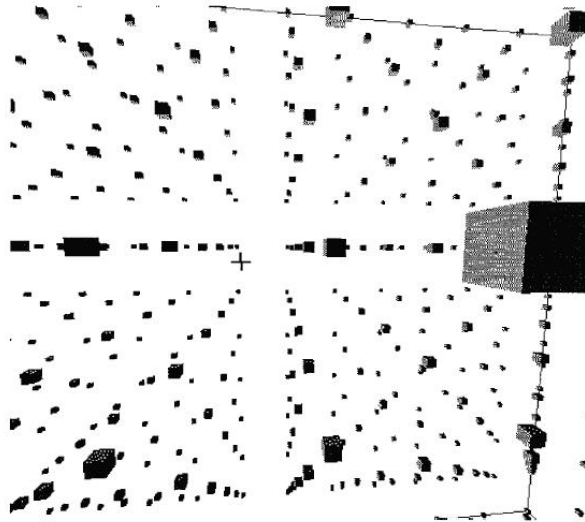


FIGURE 2.8 – Le curseur est placé dans chacun des cubes, **GHAZISAEDY et collab.** [1995].

L'utilisateur manipule un curseur qui indique la position du capteur dans le volume de mesure. Le capteur est fixé à l'extrémité du capteur ultrasonique. Il porte des lunettes stéréoscopiques qui lui permettent de voir le curseur. Ensuite, il positionne le curseur jusqu'à ce qu'il le juge au centre du cube (figure 2.8). La différence de position entre les deux systèmes de tracking est l'erreur engendrée essentiellement par les distorsions électromagnétiques.

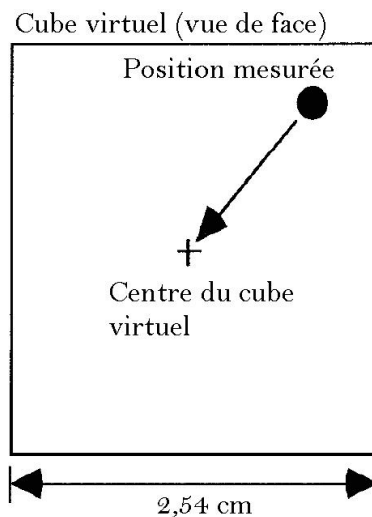


FIGURE 2.9 – Les points collectés sont décalés d'une distance par rapport au centre du cube, **GHAZISAEDY et collab.** [1995].

L'utilisateur répète cette procédure jusqu'à ce que tous les cubes à l'intérieur du volume de mesure soient pris en considération. Le système tracking à ultrasons a les avantages d'avoir une bonne linéarité pour les grandes distances, d'avoir une précision meilleure que du tracking magnétique, d'être insensible aux champs magnétiques et d'être relativement peu coûteux. Cependant, il ne permet pas la mesure de l'orientation et nécessite de maintenir les transducteurs parallèles aux murs. De plus, le chemin parcouru par le signal ne doit pas être physiquement bloqué. Cette technique d'acquisition est meilleure pour corriger les erreurs de grandes amplitudes plutôt que les petites.

LIVINGSTON et STATE [1997] ont utilisé un système de tracking mécanique, Faro Metrecom IND-1, pour déterminer la position et l'orientation du capteur électromagnétique de type Flock of Birds dans l'espace vrai. Le capteur magnétique est monté de façon rigide à l'extrémité d'un bras de Faro par une tige en plastique articulée.

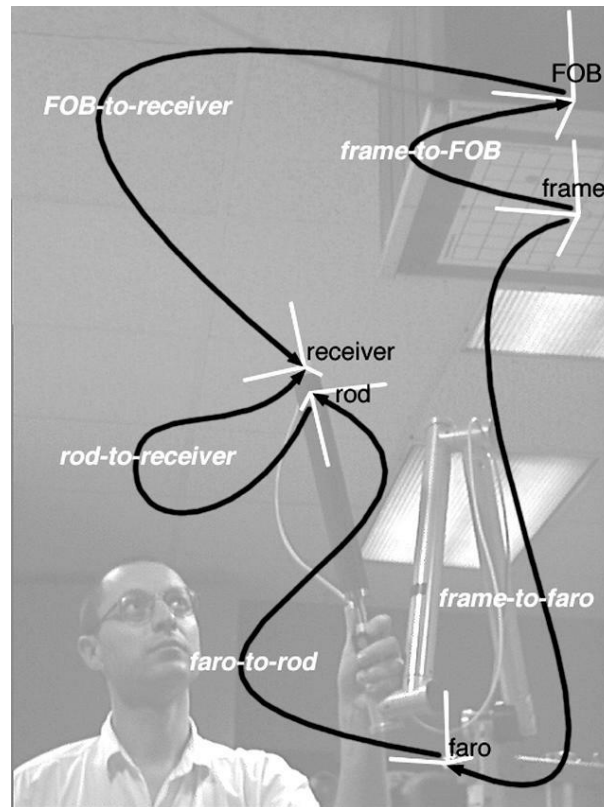


FIGURE 2.10 – Montage expérimental pour la collecte des mesures utilisant un bras de FARO, LIVINGSTON et STATE [1997].

Le Faro est un dispositif mécanique à six degrés de liberté dont le rayon d'action est de $1,83m$. Les deux séries de mesures p_i et q_i sont enregistrées au même temps. Les mesures ont été collectées aléatoirement à l'intérieur du volume de travail puis ré-échantillonnées de manière à former une mire à trois dimensions rectilinéaire. La procédure de ré-échantillonnage utilise un noyau gaussien centré autour de chaque nœud de la mire avec un rayon de coupure de $15cm$. Pour chaque échantillon, plusieurs lectures ont été faites pour réduire l'effet du bruit sur la mesure. Onze enregistrements ont été pris avec le système de tracking électromagnétique et trois mesures avec le faro. La moyenne des enregistrements a été utilisée pour les calculs. Un total de 12801 échantillons a été conservé. En effet, les échantillons restants appartenaient à une sphère de deux mètres. Pour évaluer les performances de la technique de calibration, 5334 échantillons ont été extraits du premier nuage de points. Les mesures qui avaient des grandes erreurs ont été éliminées des calculs (1787 des 12801 mesures). La collecte des mesures de référence avec le bras de Faro peut être très rapide et précis et l'orientation des capteurs électromagnétiques peut être aussi arbitraire. En revanche, le bras mécanique de Faro a un coût élevé.

BIRKFELLNER et collab. [1998] ont développé une technique d'acquisition des données similaires à celle proposée dans LIVINGSTON et STATE [1997]. Un autre protocole a été conçu pour éliminer les problèmes d'obstruction rencontrés lors des séances de chirurgie assistée par ordinateur à l'aide d'un système de tracking optique. Les équipements utilisés comportent un système de tracking optique Flashpoint 5000 de l'entreprise Image

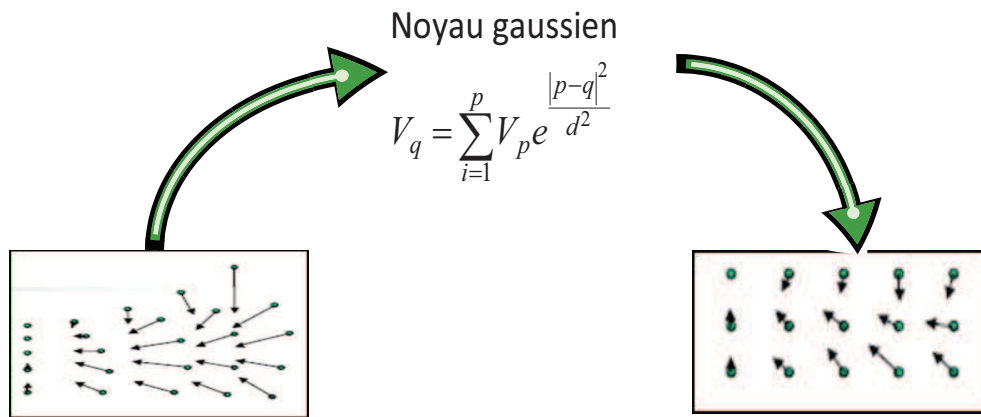


FIGURE 2.11 – La procédure de ré-échantillonnage.

Guided Technologies et un système de tracking magnétique "The Bird" de l'entreprise Accension Technologies Inc. Le dispositif de calibration est un stylet de fibre de carbone et de titane, conçu pour permettre de monter à la fois le capteur électromagnétique et seize diodes électroluminescentes du système de tracking optique. Les mesures issues de deux systèmes sont filtrées : une mesure est rejetée si l'écart de position entre deux mesures prises par le capteur est supérieur à l'exactitude garantie par le fabricant.

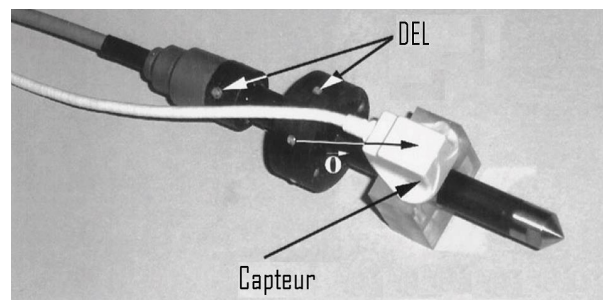


FIGURE 2.12 – Le stylet de mesure hybride utilisé dans l'expérimentation, [BIRKFELLNER et collab. \[1998\]](#).

Les expériences ont été réalisées dans une salle de chirurgie buccale et maxillo-faciale dont la majorité des pièces de la table et des instruments de la salle d'opération sont fabriqués d'acier austénitique non ferromagnétique (figure 2.13). L'émetteur du système de tracking électromagnétique a été installé en dessous de la table d'opération en raison du rayon d'action relativement petit ($0,915m$) de l'émetteur. Le volume de mesure a les dimensions $0,35m \times 0,35m \times 0,35m$ et la résolution utilisée est de l'ordre de cinq centimètres. Cinq séries de mesures ont été effectués pour déterminer le paramètre initial de la calibration du stylet de mesure hybride. Ce paramètre est la distance entre la diode électroluminescente de référence du stylet et le point de mesure. Pour chaque mesure, dix enregistrements ont été utilisés à l'intérieur du volume de mesure. Pour chaque mesure, un couple de coordonnées provenant des deux systèmes de tracking est enregistré. L'orientation et la hauteur de la table d'opération sont rectifiées pour chaque mesure, de même que la position du système de tracking optique dans la salle de chirurgie.

La table de correspondance est remplie en déplaçant le stylet à cinq cents douze emplacements différents à l'intérieur du volume de mesure et pour lesquelles l'orientation du stylet est variable. La qualité de la calibration de l'erreur dépend cependant du nombre de mesures à l'intérieur du volume de mesure. Les résultats obtenus indiquent qu'il n'est possible d'enregistrer les mesures de calibration qu'une seule fois, même si le volume

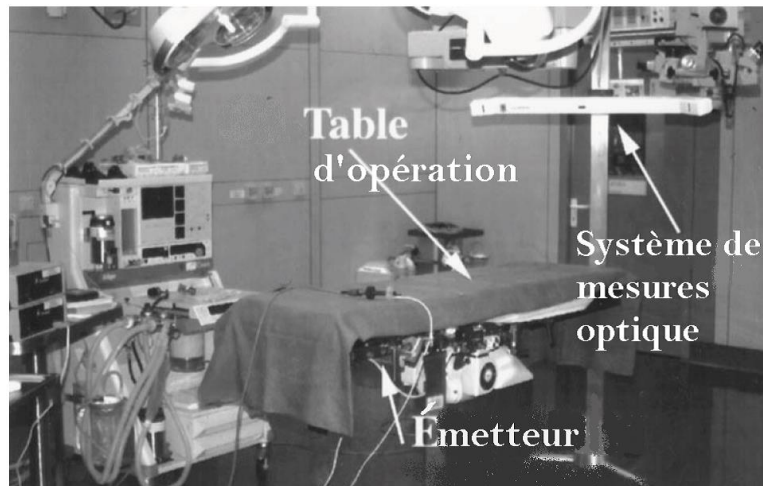


FIGURE 2.13 – Salle de chirurgie utilisée pour les expérimentations, **BIRKFELLNER et collab.** [1998].

d'utilisation (salle de chirurgie) change entre deux séances d'utilisation. La salle de chirurgie est composée essentiellement de pièces non ferromagnétiques contribue à la stabilité des paramètres de la table de correspondances.

IKITS et collab. [2001] ont utilisé un système de tracking optique de type NDI Optotrack 3020 avec quatre marqueurs pour mesurer la position de référence du capteur électromagnétique dans l'espace vari Q (figure 2.14). Le système développé utilise des marqueurs infrarouges et des équipements vidéo spécialisés pour déterminer la position du capteur électromagnétique (Polhemus Fastrak). Le capteur électromagnétique et les marqueurs infrarouges sont attachés à une plateforme non métallique souple qui peut être déplacée et obtenir une orientation quelconque (six degrés de liberté). La collecte des mesures par les systèmes s'effectue en même temps ce qui permet de déterminer la position et l'orientation du capteur dans les deux repères de suivi P et vrai Q. Un total de six cents mesures a été enregistré à l'intérieur d'un volume de $0,9m \times 1,5m \times 1,8m$. De ce nombre, 480 mesures ont servi pour la calibration et 120 mesures prises aléatoirement pour vérifier la technique en question.

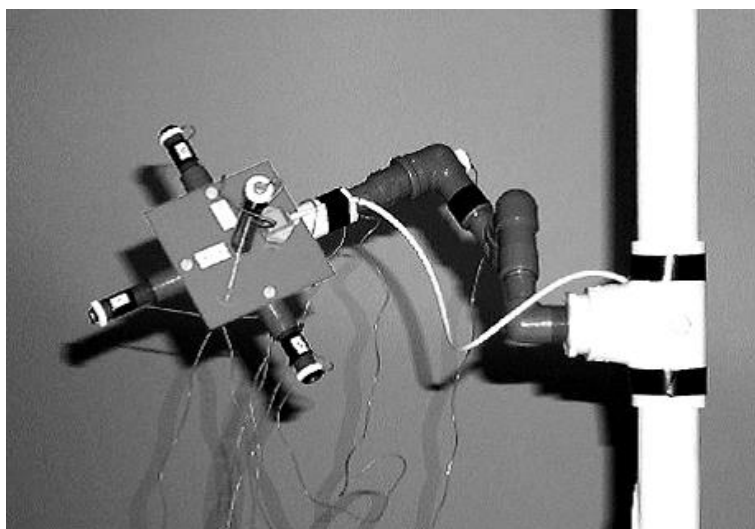


FIGURE 2.14 – Plateforme flexible utilisée pour fixer les capteurs et les marqueurs optiques, **IKITS et collab.** [2001].

Une résolution de $0,15cm$ a été adoptée, ce qui implique aussi que certaines mesures

ont été exclues. L'orientation du capteur a été randomisée durant la calibration pour couvrir l'ensemble des rotations possibles. A chaque point de mesure, cent vingt d'enregistrements ont été recueillies par le capteur électromagnétique et cent par le système de tracking optique Optotrak. La moyenne des enregistrements a été ensuite utilisée dans les calculs. La source du système de tracking électromagnétique était située à $0,8m$ du sol.

Dans d'autres travaux, PÉRIÉ et collab. [2002] ont évalué l'utilisation d'un émetteur à grande portée pour des applications de biomécanique spécifique à un environnement qui comprend une plateforme de force. Les équipements utilisés comprennent un système de tracking électromagnétique de type Motion Star de l'entreprise Ascension, un montage d'acquisition des données similaire à celui de Day et al, un système de tracking optique et une plateforme de force (OR6-1 AMTI). Le volume de mesure est de $1,4m \times 0,8m \times 1,2m$. La résolution employée est de l'ordre de vingt centimètres. La plateforme de force de dimensions $0,46m \times 0,51m \times 0,08m^3$ est placée au centre du volume. L'émetteur est centré à $1,15m$ au-dessus du sol et à $0,75m$ du site de mesure le plus proche (figure 2.15). Le montage permet d'enregistrer les coordonnées de 280 mesures. A chaque mesure, quatre différentes mesures d'orientation prédéterminée sont enregistrées. L'écart-type des mesures est calculé avec 100 mesures. La reproductibilité est validée par deux séries d'enregistrement qui utilisent les mêmes points de mesures et deux autres séries d'enregistrement avec des capteurs différents.

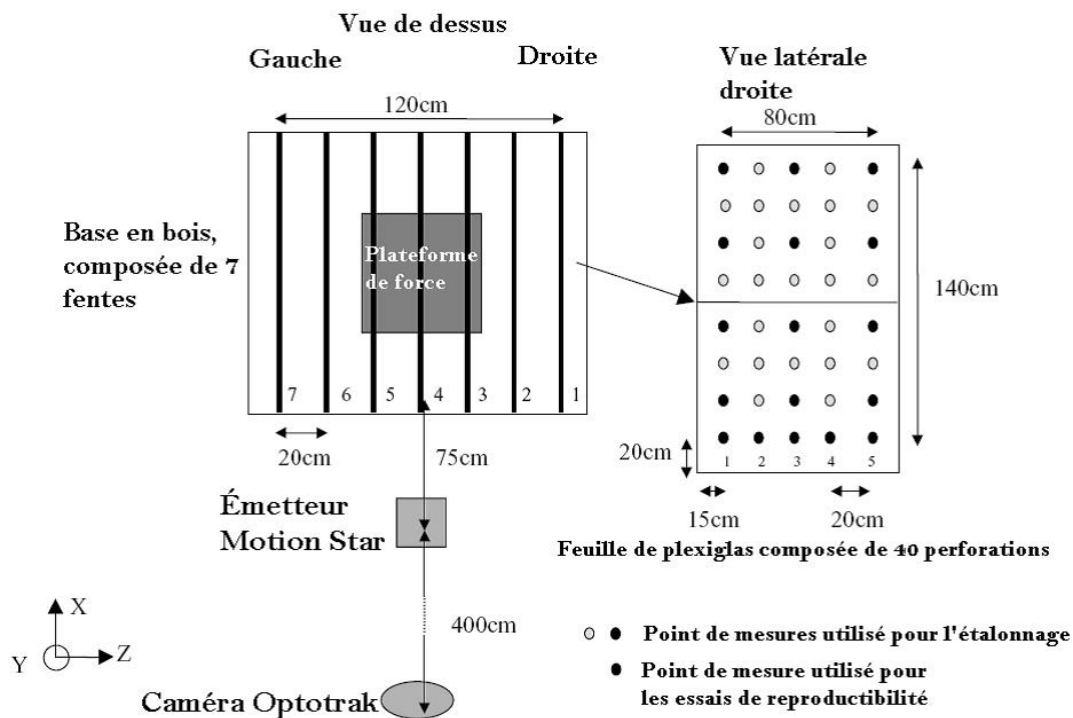


FIGURE 2.15 – Montage de calibration : la mire est composée de 40 trous espacés de $20cm$, PÉRIÉ et collab. [2002].

FRANTZ et collab. [2003] ont développé un protocole volumétrique permettant d'évaluer les performances d'un système de tracking magnétique sur le volume opérationnel entier. Il peut être aussi un guide utile pour sélectionner la région et l'orientation à utiliser dans les applications qui ne requièrent pas tous l'usage de la totalité du volume. Ils ont fait appel à un bras de robot pour déplacer un certain nombre de capteurs magnétiques à cinq degrés de liberté d'une manière aléatoire dans le volume de mesure. Un grand nombre de mesures de position et d'orientation a été prélevé pour garantir un échantillonnage re-

présentatif de l'espace. Le protocole est illustré par la figure 2.16. Pour chaque mesure de position ou d'orientation, soixante échantillons de données brutes ont été enregistrées. Chaque paire mesurée a été également mesurée par un système de tracking de référence (un système de tracking optique de type Optrak). Les mesures de références ont une précision nettement meilleure que le système de tracking magnétique. Les deux ensembles de mesure ont été ensuite alignés pour évaluer la précision en comparant les mesures de tracking magnétique à celles retournées par le tracking optique de référence.

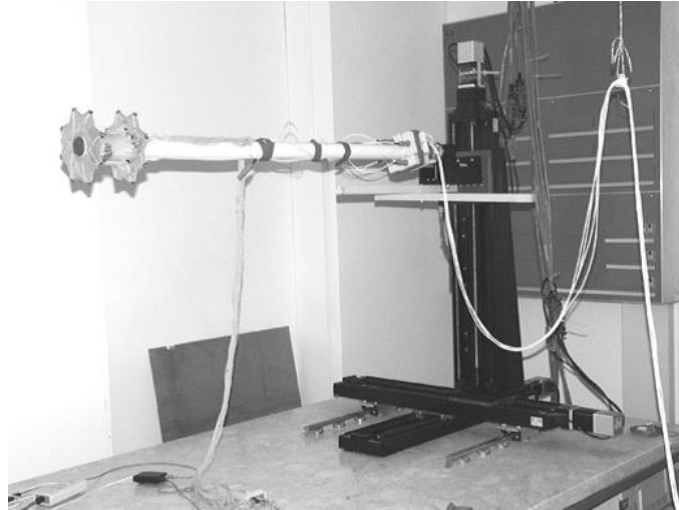


FIGURE 2.16 – Montage utilisé dans le protocole de calibration volumique , [FRANTZ et collab. \[2003\]](#).

[WU et TAYLOR \[2003\]](#) ont utilisé un système de tracking optique de type Optrak 3020 comme référence. Les marqueurs actifs (micro LED) de l'Optrak sont fixés sur le générateur du champ magnétique Aurora pour effectuer les transformations nécessaires entre les deux systèmes de coordonnées. L'objet de calibration est composé d'un bloc rectangulaire en plastique avec des motifs de rainures usinés de chaque face. Les capteurs électromagnétiques ont été logés dans les rainures réparties d'une manière à réduire les interférences entre les capteurs. L'objet est attaché à un petit robot Lego en plastique pour pouvoir le déplacer dans l'environnement de travail. Un volume d'environ 200 mm^3 situé devant le générateur de champ magnétique a été calibré.

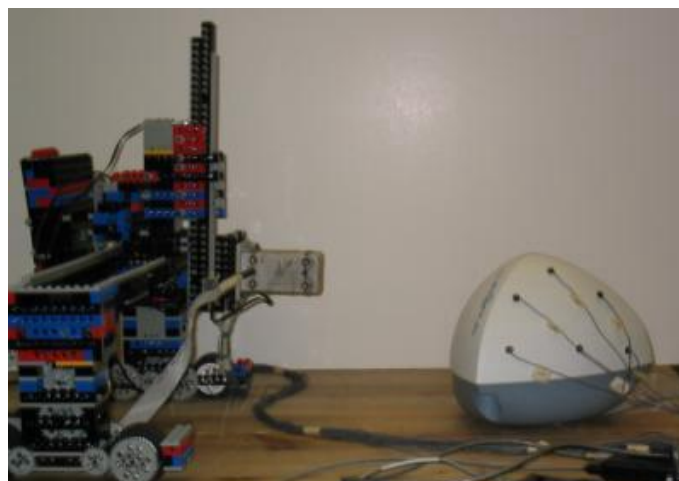


FIGURE 2.17 – Un robot Lego en plastique pour la collecte de mesures , [WU et TAYLOR \[2003\]](#).

Récemment, MOREIRA et collab. [2014] ont développé un nouveau protocole pour la calibration des systèmes de tracking électromagnétiques utilisés dans la conception des superstructures pour les patients édentés.

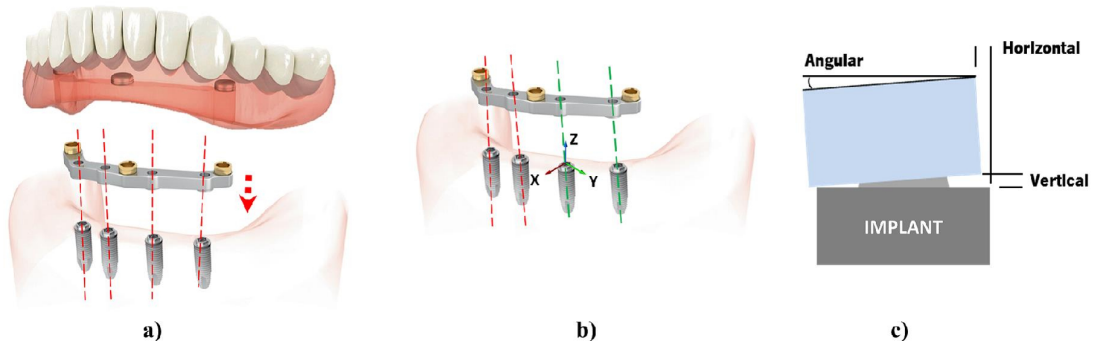


FIGURE 2.18 – Exemples d’implants et prothèses dentaires, MOREIRA et collab. [2014].

Le protocole proposé a abouti au développement d’un objet miniature qui combine les implants dentaires et les capteurs électromagnétiques. Dans le but d’évaluer la fiabilité des mesures de position et d’orientation des implants dentaires retournée par le système de tracking électromagnétique, une calibration a été effectuée au laboratoire. Un robot industriel de six degrés de liberté (ABB IRB 140 avec une reproductibilité de $30\mu m$) a été utilisé pour manipuler l’objet conçu dans un volume de $300x300x300mm$. L’objet a été fixé à l’extrémité d’un tube en fibre de carbone de section $10x10mm$ avec une longueur de $600mm$ et de l’autre côté au pointe du robot. Ce système a été utilisé pour l’acquisition des échantillons au sein du volume mentionné précédemment avec une résolution de 30 micromètres qui vont être prises comme des mesures de référence dans la procédure de calibration. Pour chaque point, mille échantillon de système de tracking magnétique ont été acquises sans attacher l’implant la pointe de l’objet. En complément de ces points, 30 autres points de validation ont été récoltés pour évaluer la performance de la calibration.

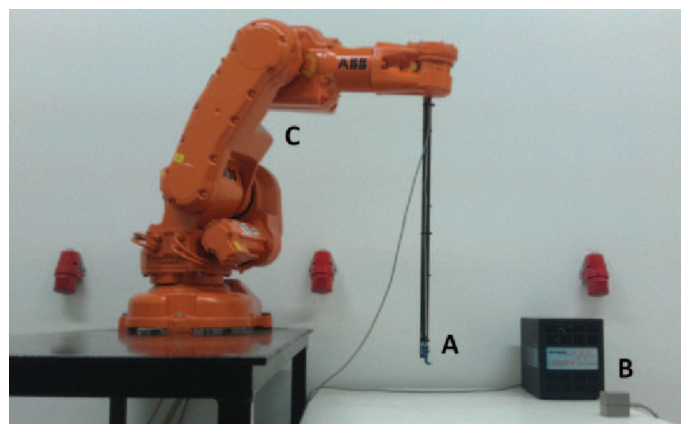


FIGURE 2.19 – Configuration du système : (A) L’objet, (B) Le système de tracking magnétique Polhemus et (C) le robot ABB IRB 140, MOREIRA et collab. [2014].

2.2 CLASSIFICATION DES PROTOCOLES DE COLLECTE DE DONNÉES

Pour représenter la distribution des erreurs de mesures d'un système de tracking, un protocole de collecte de données doit être choisi. Dans le présent paragraphe, nous confrontons les différents protocoles afin de dégager les avantages et les inconvénients.

2.2.1 Collecte de données dynamique versus statique

La collecte de données dynamique consiste à recueillir les données comme étant un système hybride, un objet rigide combinant les deux systèmes de tracking est déplacé dans le volume de travail sans s'arrêter pour prendre les données des deux systèmes. Cette méthode semble idéale car elle permet de collecter une énorme quantité de données dans une courte période de temps. Néanmoins, plusieurs problèmes se posent : diminution de précision de tracking des objets mobiles, la latence entre les lectures et les mesures réelles, la nécessité d'une calibration temporelle entre les systèmes de tracking, une réparation de données hétérogène (dans le cas d'une collecte manuelle). Il est possible de remédier aux problèmes deux et trois en faisant une calibration temporelle qui mesure l'intervalle entre les moments où les mesures sont disponibles pour chaque système. Des travaux antérieurs de [WU et TAYLOR \[2003\]](#) et [NAKADA et collab. \[2003\]](#) décrivent les méthodes possibles pour effectuer une telle calibration temporelle et plus précisément comment réduire ce décalage temporel. Cependant, il s'est avéré que cette méthode n'est pas idéale, pour cela une méthode quasi statique a été proposée. L'objet est placé à une position donnée. Une lecture est effectuée par les deux systèmes puis il est déplacé vers une autre position. Il s'agit d'un scénario idéal pour la collecte de données pour les systèmes robotisés.

2.2.2 Collecte de données automatique versus manuelle

Les données peuvent être collectées par le biais des systèmes robotisés ou complètement manuels. Nous distinguons deux types de collecte manuelle : "Freehand" où un objet est déplacé dans le volume et une acquisition de données guidée par une mire de calibration. La première méthode est rapide et permet de recueillir une grande quantité de données mais elle est sujette à une moindre précision vu sa nature dynamique et conduira à une répartition hétérogène des données. La seconde méthode, fournit une bonne distribution de données qui peuvent être collectées d'une façon quasi-statique mais elle est extrêmement fastidieuse.

D'autre part, la collection de données automatique est idéale pour rassembler un grand nombre de données avec une distribution prédéterminée en utilisant un mode quasi statique de collecte et demande une interaction minimale de la part de l'utilisateur. Les inconvénients de la collection basées sur les robots est qu'elle nécessite des équipements à savoir un robot qui ne soit pas tout le temps pratique dans le cas de collecte de données dans un contexte de réalité virtuelle.

2.3 CLASSIFICATION DES ERREURS

Les sources d'erreur en tracking et capture de mouvement peuvent être classées généralement en deux grandes catégories : la première comprend toutes les erreurs liées à la réalisation des mesures statiques soit hors ligne avant de se lancer dans l'application, ou en ligne pendant le fonctionnement normal. Nous les appelons les erreurs de mesure statique. La seconde comprend toutes les erreurs qui découlent des inévitables sources de retard dans le pipeline de tracking ALLEN et collab. [2001] ou de la dynamique du mouvement.

Un aperçu sur les différentes classes d'erreurs de tracking est donné par la figure 2.20 et qui seront décrites en détails dans les sections 2.3.1 et 2.3.2.

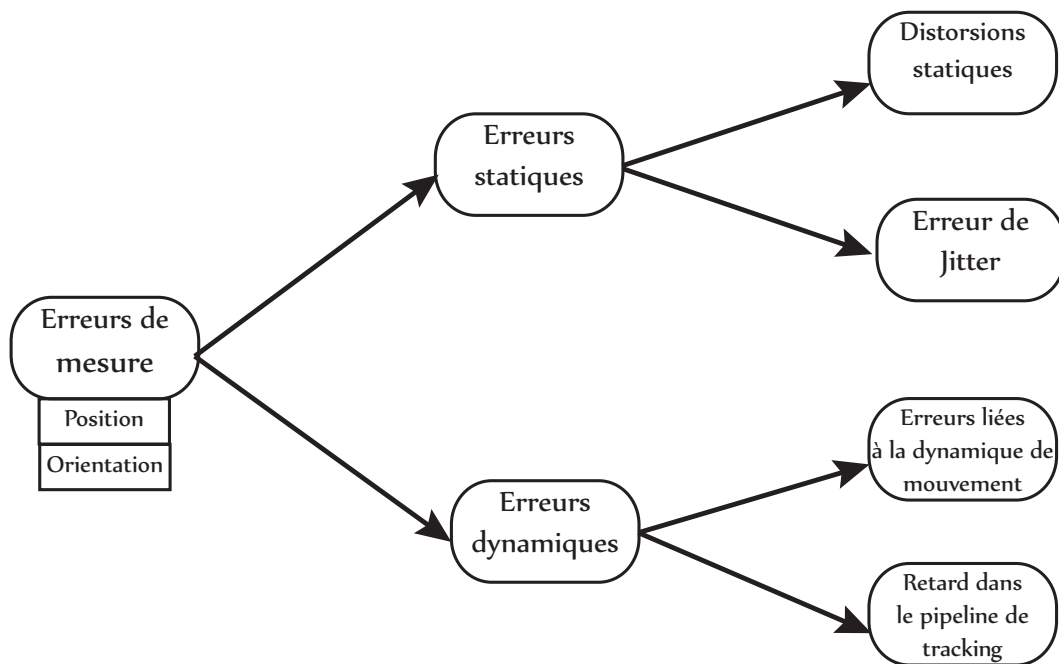


FIGURE 2.20 – Classification des erreurs de tracking.

2.3.1 Les erreurs statiques

2.3.1.1 Distorsions statiques

Pour les systèmes de tracking, nous divisons les erreurs statiques en deux types : répétable et non répétable. Certains systèmes de tracking ont des erreurs systématiques dans leur volume de mesure qui causent des erreurs de grande valeur. Nous les appelons distorsions du champ statique. Ces erreurs de mesure sont répétables ce qui permet de les caractériser et les corriger tant qu'elles restent constantes entre la procédure de calibration et le moment d'exécution.

2.3.1.2 Les erreurs aléatoires ou Gigue (Jitter)

Nous considérons les erreurs non répétables induites par le système de tracking lorsque l'objet à traquer est immobile. Ce type d'erreur est non répétable et donc ne peut pas être

corrigé à priori via la calibration. La gigue est souvent proportionnelle à la distance entre les objets à suivre et les sources et augmente aux frontières du volume de travail.

2.3.2 Les erreurs dynamiques

Toute mesure d'un phénomène variable dans le temps non répétable est valide au mieux à l'instant où l'échantillon se produit puis devient usée avec le passage du temps jusqu'à l'arrivée du prochain échantillon. L'âge de données est par conséquent un facteur important dans sa précision. Tout retard entre le moment où la mesure s'est produite et le moment où elle est consommée par le système, pour estimer la pose, contribue à l'augmentation de l'âge de l'information et donc à l'imprécision de la mesure. Les erreurs dynamiques sont moins évidentes quand elles se produisent. Dans la suite nous allons nous concentrer principalement sur l'erreur dynamique de premier ordre. L'effet le plus évident est l'erreur dynamique provoquée par le mouvement continu de l'utilisateur après qu'un cycle de tracking a déjà commencé. Si la main ou la tête de l'utilisateur est en rotation avec une vitesse angulaire $\dot{\theta}$ et en translation avec une vitesse linéaire \dot{x} . L'erreur dynamique de premier ordre s'écrit alors :

$$\begin{aligned}\varepsilon_{dyn,\theta} &= \dot{\theta} \Delta t \\ \varepsilon_{dyn,x} &= \dot{x} \Delta t\end{aligned}\tag{2.1}$$

Où $\Delta t = \Delta t_m + \Delta t_g$ désigne la somme du retard de mouvement total Δt_m du système de tracking, et de Δt_g , le retard à travers le reste du pipeline graphique, y compris le rendu et la génération des graphiques, le retard introduit par la synchronisation vidéo etc. . .

Bien qu'il existe certainement d'autres latences dans un système typique de réalité virtuelle, la latence de tracking est la seule qui détermine combien de temps s'est écoulé avant la première occasion de répondre au mouvement de l'utilisateur. En effet, quand l'utilisateur bouge, nous souhaitons le connaître le plus tôt possible. Le pipeline du tracking comporte à la fois des latences fixes associées à des tâches bien connues telles que l'exécution des fonctions qui calculent la position et des latences variables associées à la synchronisation entre des tâches asynchrones bien définies. Ce dernier est souvent appelé retard de synchronisation ou encore rendez-vous retard. Dans l'exemple de la figure 2.21, les mesures et l'estimation de la position se produisent à des intervalles réguliers mais à des fréquences différentes. Inévitablement, toute mesure durera un certain temps avant d'être utilisé pour calculer une estimation de la position.

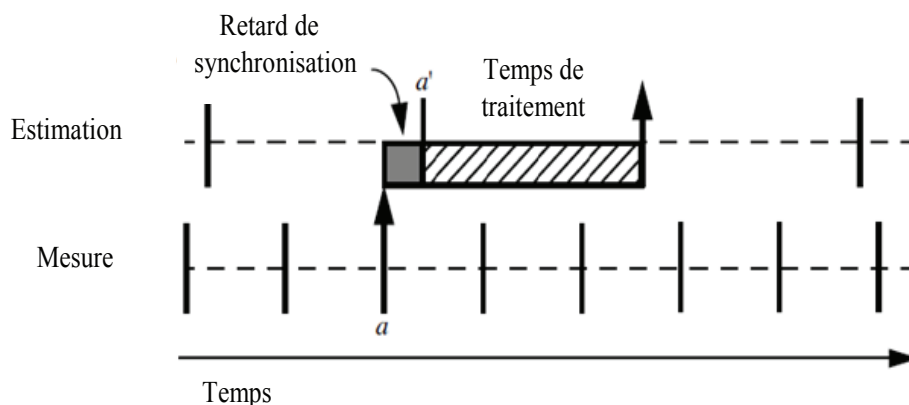


FIGURE 2.21 – Retard de synchronisation (Synchronisation delay).

2.3.3 L'erreur totale de tracking

L'erreur totale s'obtient en sommant l'erreur statique à l'erreur dynamique de premier ordre.

$$\begin{aligned}\varepsilon_{\theta} &= \varepsilon_{stat,\theta} + \varepsilon_{dyn,\theta} \\ \varepsilon_x &= \varepsilon_{stat,x} + \varepsilon_{dyn,x}\end{aligned}\tag{2.2}$$

Où ε_{stat} , ε_{dyn} désignent respectivement l'erreur statique et dynamique de premier ordre.

2.4 LES TECHNIQUES DE CALIBRATION

La plupart des techniques de calibration des erreurs de position et d'orientation présentées dans cette section sont prises de la revue de la littérature effectuée par **KINDRA-TENKO [2000]**. En effet, les systèmes de tracking magnétique sont reconnus comme étant les systèmes les plus employés pour les applications de réalité virtuelle étant donné leurs faibles coûts et leurs larges étendues de mesure.

La procédure commune pour la correction des erreurs statiques, pour les systèmes de tracking électromagnétiques consiste à enregistrer un ensemble de mesures entachées d'erreur ainsi que les mesures de référence correspondantes avec un système de tracking secondaire ou un support physique ou virtuel. La procédure de calibration des grandes erreurs statiques comprend trois étapes :

- Recueillir n paires de mesures de position (p_i, q_i) correspondantes aux positions retournées par le système de "tracking" $\{p_i \in P | i = 1, \dots, N\}$ et aux positions correctes $\{q_i \in Q | i = 1, \dots, N\}$ retournées par un système de "tracking" secondaire plus précis.
- Rechercher la fonction f permettant de mettre en relation les deux séries de mesures : $f : P \rightarrow Q$ tel que $f(p_i) = q_i \forall i = 1, \dots, N$
- Appliquer f à chaque donnée renvoyée par le système de tracking en temps réel (calculer $f(p_{courante})$) et estimer la position corrigée courante.

Les coordonnées des mesures sont exprimées comme suit :

$$\begin{aligned}p_i &= (x_i^p, y_i^p, z_i^p) \\ q_i &= (x_i^q, y_i^q, z_i^q)\end{aligned}$$

Les techniques de calibration des erreurs statiques peuvent être classées en deux catégories : méthodes locales et globales. Les approches locales divisent le volume de travail en petits sous espaces et définit une unique relation pour chaque sous espace. Contrairement, les techniques globales offrent une seule fonction pour tout l'espace de tracking.

2.4.1 L'interpolation tri-linéaire

L'interpolation tri-linéaire effectue une correspondance unique entre n'importe quelle mesure à l'intérieur du volume d'un cube (espace de suivi P) et une coordonnée de position à l'intérieur d'un polyèdre quelconque dans l'espace vrai Q. L'étalonnage s'effectue

d'abord en prenant huit mesures $\{p_1, \dots, p_8\}$ de manière à former un cube à l'intérieur de l'espace P.

Ensuite, il suffit de déterminer les coordonnées vraies $\{q_1, \dots, q_8\}$. Cette technique ne fonctionne correctement uniquement que si le volume d'étalonnage est entièrement subdivisé en petits cubes adjacents les uns des autres. Les subdivisions sont identifiées par les coordonnées $\{p_i\}_j, i = 1, \dots, 8, j = 1, \dots, m$ où m est le nombre de subdivisions dans l'espace de suivi P. Une table de mise en correspondance est utilisée pour localiser n'importe quel point de mesure p_i à l'intérieur du volume P via l'un des cubes $\{p_i\}_j$ formés préalablement. Les équations (2.3) et (2.4) sont utilisées pour faire correspondre une mesure p_i vers sa position correspondante dans l'espace vrai [PRESS et collab. \[1987\]](#).

$$\begin{aligned}
 q = & (1-t)(1-u)(1-w)q_1 \\
 & + t(1-u)(1-w)q_2 + tu(1-w)q_3 \\
 & + (1-t)u(1-w)q_4 + (1-t)(1-u)q_5 \\
 & + t(1-t)wq_6 + (1-t)wq_7 + tuwq_8
 \end{aligned} \tag{2.3}$$

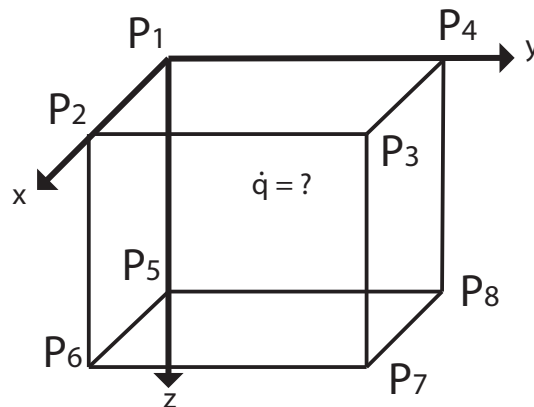


FIGURE 2.22 – L'interpolation tri linéaire.

$$t = \frac{x^p - x_1^p}{x^p - x_2^p}, u = \frac{y^p - y_1^p}{y^p - y_2^p}, w = \frac{z^p - z_1^p}{z^p - z_2^p} \tag{2.4}$$

La technique d'interpolation tri-linéaire fournit un suivi continu de classe C^1 (une fonction f est de classe C^1 si elle est dérivable et sa dérivé est continue) à l'intérieur de chaque cube enregistrée. Néanmoins, les gradients de la fonction d'interpolation changent rapidement à la frontière des cubes entrecroisés. A ces endroits, la fonction d'interpolation subit des discontinuités. En outre, l'acquisition des mesures d'une mire rectilinéaire dans l'espace de suivi (entaché d'erreur) est un problème non-trivial et fait généralement recours à un ré-échantillonnage. En effet, cette technique impose par le fait même une restriction lors de la collecte des points de mesures. Chaque petite subdivision doit être formée de huit mesures formant un cube. Plus les cubes seront petits et plus la précision sera augmentée. Plusieurs auteurs ont retenu cette technique dans leurs recherches dont : [GHAZISAEDY et collab. \[1995\]](#), [LIVINGSTON et STATE \[1997\]](#), [CZERNUSZENKO \[1998\]](#) et [MOREIRA et collab. \[2014\]](#).

2.4.2 L'interpolation par le point le plus proche

Concernant cette technique de compensation de l'erreur de position, un capteur est déplacé à huit positions $\{q_1, \dots, q_8\}$ différentes de façon à former un cube dans l'espace vrai Q. Les points $\{p_1, \dots, p_8\}$ de mesures sont inscrits dans une table de correspondances. Les huit points de mesures forment ainsi un polyèdre dans l'espace P. L'erreur de position $\{v_i\}$ de la mesure $\{p_i\}$ est ensuite calculée avec l'équation (2.5)

$$v_i = q_i - p_i \quad (2.5)$$

BRYSON [1992] et KINDRATENKO [2000] ont utilisé des techniques qui nécessitent que l'ensemble de l'espace Q soit segmenté en cubes dont les sommets forment les références $\{q_i\}_j$. Pour chacune de ces subdivisions, les mesures $\{p_i\}_j$ correspondantes sont enregistrées. C'est-à-dire que pour toutes les mesures $\forall p \in P$, leurs polyèdres $\{p_i\}_j$ associés sont situés dans une table de correspondance. Par la suite, l'image q d'une mesure p peut être retrouvée à l'intérieur de l'espace réel Q par l'équation (2.5) et (2.6).

BRYSON [1992] et BRIGGS [1999] déterminent le point q correspondant au point p situé à l'intérieur du polyèdre défini par $\{p_1, \dots, p_8\}$ par l'équation (2.6) :

$$q = p + \sum_{i=1}^8 w_i v_i \quad (2.6)$$

Une pondération particulière $\{w_i\}$ influence le choix du point q , de telle sorte à ce que la somme des coefficients de pondération de l'équation (2.5) soit égale à 1 :

$$w_1 + \dots + w_8 = 1 \quad (2.7)$$

A l'équation (2.8), quand toutes les valeurs de la variable r (2.9), (2.10), (2.14), (2.15), (2.16) sont égales entre elles c'est à dire $r_1 = \dots = r_8$, la somme des erreurs pondérées $\sum_{i=1}^8 w_i v_i$ est égale à la moyenne des huit erreurs $\sum_{i=1}^8 \frac{v_i}{8}$

$$w_i = \frac{r_i}{\sum_{i=1}^8 \frac{v_i}{8}} \quad (2.8)$$

2.4.2.1 Méthode de pondération linéaire

Dans la méthode de pondération linéaire, le poids relatif est défini comme une fonction de la distance d (2.9).

$$r_i = \prod_{j=1, j \neq i}^8 d_j \quad (2.9)$$

2.4.2.2 Méthode de pondération exponentielle

La méthode de pondération exponentielle (2.10) quant-à elle, caractérise le poids relatif comme une fonction exponentielle décroissante de la distance entre p et p_i .

$$r_i = \begin{cases} e^{-d_i} & \text{si } d_i < d_{\max} \\ 0 & \text{si } d_i \geq d_{\max} \end{cases} \quad (2.10)$$

Dans l'expression mentionnée ci-dessus, la variable d_i est la distance entre les points p et p_i . La variable d_{max} est une constante qui représente une distance, qui est plus petite que la plus petite des distances procurées par les coordonnées des mesures $\{p_1, \dots, p_8\}$.

2.4.2.3 Méthode de pondération basée sur le carré de la distance

La méthode de pondération basée sur le carré de la distance pondère l'erreur de position en fonction du carré de la distance séparant les mesures p et p_i . Elle nécessite des mesures p_i et q_i identiques à la topologie utilisée par technique d'interpolation tri linéaire. Par conséquent, les mesures p_i forment un cube dans l'espace P et les mesures q_i un polyèdre associé à l'espace Q. Pour chaque mesure $\{p_i\}$ l'erreur de position vectorielle est déterminée par l'équation (2.11).

$$v_i = p_i - q_i \quad (2.11)$$

L'équation de correction de l'erreur de mesure est représentée par l'équation (2.12)

$$q = p - \sum_{i=1}^8 w_i v_i \quad (2.12)$$

La fonction de pondération de l'erreur est définie par l'équation (2.13), où la variable d_i est la distance entre les mesures p et p_i .

$$w_i = \frac{1}{d_i^2} \sum_{i=1}^8 d_i^2 \quad (2.13)$$

2.4.2.4 Autres méthodes de pondération

KINDRATENKO et BENNETT [2000] proposent trois autres différentes fonctions pondérées (2.14), (2.15) et (2.16).

$$r_i = \begin{cases} -\ln \frac{d_i}{d_{max}} & \text{si } d_i < d_{max} \\ 0 & \text{si } d_i \geq d_{max} \end{cases} \quad (2.14)$$

$$r_i = \begin{cases} 1 - \frac{d_i}{d_{max}} & \text{si } d_i < d_{max} \\ 0 & \text{si } d_i \geq d_{max} \end{cases} \quad (2.15)$$

$$r_i = \begin{cases} 1 - \frac{d_i^2}{d_{max}^2} & \text{si } d_i < d_{max} \\ 0 & \text{si } d_i \geq d_{max} \end{cases} \quad (2.16)$$

2.4.3 Interpolation polynomiale

La technique de calibration par polynômes permet également de compenser les erreurs de position BRYSON [1992], KINDRATENKO [1999], IKITS et collab. [2001], NAKADA et collab. [2003] et d'orientation KINDRATENKO et BENNETT [2000], IKITS et collab. [2001], NIXON et collab. [1998]. Dans cette technique, des mesures q_i sont effectuées à différents endroits dans une mire rectilinéaire de l'espace Q, sachant que les mesures correspondantes de l'espace P sont notées p_i . Le nombre total n de mesures i du volume sont $i = 1, \dots, n$. L'erreur de position vectorielle v_i est exprimée par l'équation (2.17).

$$v_i = p_i - q_i \quad (2.17)$$

Un polynôme de degré r est utilisé pour approximer l'erreur de position en fonction des mesures $p \in P$. Il est décrit par l'équation (2.18) :

$$v = f(p) = f(x, y, z) = \sum_{j=1}^R \begin{bmatrix} c_{xj} \\ c_{yj} \\ c_{zj} \end{bmatrix} x^{s_j} y^{t_j} z^{u_j} \quad (2.18)$$

Les coefficients du polynôme sont exprimés à travers la notation suivante : c_{xj} , c_{yj} , c_{zj} . Le nombre total de termes R du polynôme doit répondre aux deux conditions suivantes (2.19) et (2.20). A la condition (2.19), les variables s_j , t_j et u_j sont tous des exposants positifs.

$$R = \frac{(r+1)(r+2)(r+3)}{6} \quad (2.19)$$

$$0 \leq s_j + t_j + u_j \leq r \quad (2.20)$$

En outre, s_j , t_j , u_j sont toutes les permutations possibles, mais de puissances uniques. Les coefficients c_{xj} , c_{yj} , c_{zj} doivent être déterminés de telle sorte à ce que la fonction polynomiale permette la meilleure approximation de l'erreur de position aux mesure p_i . Ceci est possible en minimisant la fonction coût (2.21). L'expression (2.22) peut être minimisée par la méthode des moindres carrés telle qu'énoncée par l'équation ci-dessous. Dans ce cas, $k = 1, \dots, R$ et $n > R$.

$$\sum_{i=1}^n \|v_i - f(p_i)\| \quad (2.21)$$

$$\sum_{j=1}^R \begin{bmatrix} c_{xj} \\ c_{yj} \\ c_{zj} \end{bmatrix} \cdot \sum_{i=1}^n (x^{s_j} y^{t_j} z^{u_j})(x^{s_k} y^{t_k} z^{u_k}) = \sum_{i=1}^n v_i \cdot (x^{s_k} y^{t_k} z^{u_k}) \quad (2.22)$$

La même technique peut être appliquée aussi à la compensation des erreurs de mesures en orientation. Dans ce cas, les angles d'Euler (α, β, γ) sont utilisés et exprimés par $\{o_i\}$. L'équation demeure la même sauf qu'elle est utilisée cette fois pour approximer l'erreur en orientation au lieu de l'erreur en position. Une fois que les coefficients sont calculés, les erreurs en orientation ($\alpha^e, \beta^e, \gamma^e$) pour $\forall p \in P$ peuvent être ensuite interpolées. Pour compenser l'erreur en orientation, les mesures enregistrées dans l'espace $P(\alpha, \beta, \gamma)$ et transposées dans l'espace $Q(\alpha^e, \beta^e, \gamma^e)$ doivent être converties en matrice de rotation M et M^e , en multipliant M par M^e et en reconvertissant le résultat obtenu en angles d'Euler selon **NIXON et collab. [1998]**. **IKITS et collab. [2001]** utilisent une représentation par quaternions $[w, (x, y, z)_i]$ pour la mesure d'orientation $\{o_i\}$.

L'équation (2.23) reste identique à l'exception que l'erreur de position est par la composante vectorielle (x^e, y^e, z^e) de l'erreur en orientation mise sous la forme de quaternions $[w^e, (x^e, y^e, z^e)]$. La partie scalaire du quaternion w_e est obtenue par la relation ci-après : $\|[w^e, (x^e, y^e, z^e)]\|^2 = 1$. L'erreur de mesure en orientation peut être compensée en multipliant l'inverse du quaternion interpolé par la mesure en orientation $[w, (x, y, z)_i]$ de la façon suivante :

$$[w^e, (x^e, y^e, z^e)] = [w^e, (x^e, y^e, z^e)]^{-1} \cdot [w, (x, y, z)] \quad (2.23)$$

Bien que les deux versions présentent certaines limitations de la topologie des mesures q_i , le capteur peut envisager une orientation quelconque en autant qu'elle soit connue.

La technique d'interpolation polynomiale fournit une interpolation douce et une continuité de classe C1 sur tout l'espace de suivi. Les polynômes d'ordre trois et quatre sont souvent les plus utilisés pour éviter les oscillations qui peuvent se présenter en cas d'utilisation des polynômes d'ordre supérieur. La technique permet la meilleure compensation des grandes erreurs de mesures. Cependant cette technique introduisait de petites erreurs lorsque l'erreur de mesure était initialement petite.

2.4.4 Les polynômes de Bernstein

FISCHER [2005] a utilisé les polynômes de Bernstein pour caractériser et compenser les erreurs de mesure dues aux distorsions électromagnétiques. Ils permettent une représentation globale de l'erreur et présentent des propriétés intéressantes à savoir : par construction les coefficients correspondent à l'erreur à chaque point de la mire et les dérivées sont représentées par un polynôme de Bernstein qui est généré à partir des différences entre les coefficients adjacents.

Les polynômes de Bernstein sont définis comme suit :

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i} \quad (2.24)$$

Et le coefficient binomial est :

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{si } 0 \leq i \leq n \\ 0 & \text{sinon} \end{cases} \quad (2.25)$$

Le modèle d'erreur avec les polynômes de Bernstein dans le cas simple monodimensionnel est le suivant :

$$e(x) = \sum_{i=0}^n c_i B_i^n(x) \quad (2.26)$$

Où $e(x)$ est l'erreur à la position x (x est normalisé entre 0 et 1), c_i est l' $i^{\text{ème}}$ coefficient du polynôme, $B_i^n(x)$ est le $i^{\text{ème}}$ terme de base d'un polynôme d'ordre n .

2.4.5 Technique de Hardy HMQ (Hardy's multi-quadric method)

Avec cette technique, il est possible de compenser les erreurs de position et d'orientation, et ce, même pour des mesures situées à l'extérieur du volume de calibration initial P. Cette technique est entre autre utilisée par ZACHMANN [1997] et ZACHMANN [2000]. Il n'y a pas de restriction sur la topologie des mesures. Lors de l'acquisition de ces mesures, un capteur est déplacé à différentes coordonnées q_i d'une mire rectangulaire du volume Q, sachant que les mesures correspondantes de l'espace P sont notées p_i . Le nombre total n des mesures i du volume sont $i = 1, \dots, n$. La forme générale de la fonction d'interpolation HMQ $f: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ pour les mesures de position est exprimée comme suit :

$$f(p) = \sum_{i=1}^n \alpha_j w_j(p) \quad (2.27)$$

La fonction $w_j(p)$ est décrite comme suit :

$$w_j(p) = \sqrt{\|p - p_j\|^2 + R^2} \quad (2.28)$$

Les coefficients α_i de l'équation (2.27) peuvent être calculés en résolvant un système à n équations de l'équation (2.30), en utilisant une technique de factorisation :

$$\sum_{i=1}^n \alpha_i w_j(p) = q_i \quad (2.29)$$

La régularité de l'interpolation dépend du paramètre R^2 . La valeur optimale de ce paramètre a été déterminée expérimentalement et elle devrait être comprise entre $10 < R^2 < 1000$. La valeur exacte du paramètre R^2 a un impact très faible sur la qualité de l'interpolation.

Pour compenser les mesures en orientation $\{o_i\}$, une matrice $M_{\{i\}}$ doit être utilisée. Le calcul des coefficients α_j de l'équation (2.24) se fait en résolvant l'équation suivante :

$$f(M_i^p) = M_i^q \quad (2.30)$$

Le but de la fonction HMQ est d'interpoler la mesure M^q du volume Q provenant de la mesure M^p de l'espace de calibration P. Dans ce cas, le capteur peut avoir une orientation quelconque en autant qu'elle soit connue.

L'approche de Hardy HMQ garantit une représentation continue de classe C1 sur tout l'espace de tracking et présente une interpolation très douce même aux frontières des cubes.

2.4.6 Technique basée sur les réseaux de neurones

Une technique alternative de calibration d'erreur de mesures à l'aide d'un réseau de neurones multicouches a été décrite en 2005 par KINDRATENKO et SHERMAN [2005]. Avec cette technique, il n'y a aucune restriction sur la topologie des mesures. Lors de l'acquisition de ces mesures, un capteur est déplacé à différentes coordonnées q_i d'une mire rectilinéaire de l'espace Q, sachant que les mesures correspondantes de l'espace P sont notées p_i . Le nombre total n des mesures i dans le volume est $i = 1, \dots, n$. L'ensemble des mesures récolté p_i et q_i est utilisé pour la phase d'apprentissage du réseau de neurones, de manière à ce que toutes les mesures $\forall p \in P$ aient une correspondance vers les points $q \in Q$.

KINDRATENKO et SHERMAN [2005] ont mis au point deux réseaux de neurones pour compenser les erreurs de mesures en position et en orientation. Le premier réseau consiste en trois entrées et trois sorties. Les nœuds d'entrée représentent les mesures de positions retournées par le capteur dans le volume de suivi et les sorties représentent les positions correspondantes corrigées. Une seule couche cachée placée entre les deux couches précédentes. Les auteurs ont utilisé une fonction de transfert linéaire $f(n) = n$ pour les nœuds de la couche de sortie et une fonction de transfert de type tangente sigmoïde (Tan-Sig) $f(n) = \frac{2}{(1+e^{-2n})} - 1$ pour les neurones de la couche cachée.

Un deuxième réseau de même architecture a été également aussi utilisé pour estimer les orientations corrigées. L'algorithme de Levenberg-Marquardt a été utilisé pour effectuer l'apprentissage de deux réseaux de neurones. Un ensemble de mesures de validation différent des mesures qui ont servies à l'apprentissage est utilisé pour évaluer les performances des réseaux de neurones.

La technique de calibration basée sur les réseaux de neurones a permis d'améliorer considérablement la précision du tracking. En effet, elle aboutit à des meilleurs résultats que d'autres techniques (figure 2.23) souvent utilisées pour différents systèmes de tracking électromagnétiques (Folks of Bird1, SpacePad, Folck of Birds2).

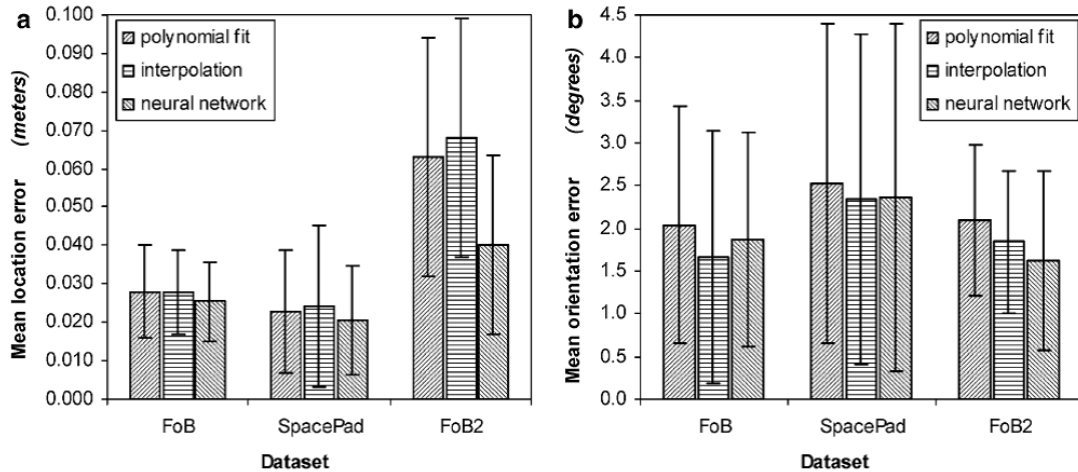


FIGURE 2.23 – Comparaison des résultats obtenus pour la position (a) et l'orientation (b). Calibration à l'aide un polynôme d'ordre 4, interpolation par le plus proche voisin et calibration basée sur les réseaux de neurones, [KINDRATENKO et SHERMAN \[2005\]](#)

Néanmoins l'inconvénient de cette approche réside dans le dimensionnement du réseau de neurones (nombre de couches cachées, nombre de neurones dans les couches cachées). L'approche adoptée par [KINDRATENKO et SHERMAN \[2005\]](#) consistait à faire l'apprentissage de plusieurs configurations de réseau de neurones en changeant à chaque fois le nombre de neurones (de 3 à 21) dans la couche cachée puis valider les performances de la configuration en question à l'aide de l'ensemble de mesures de validation (figure 2.24).

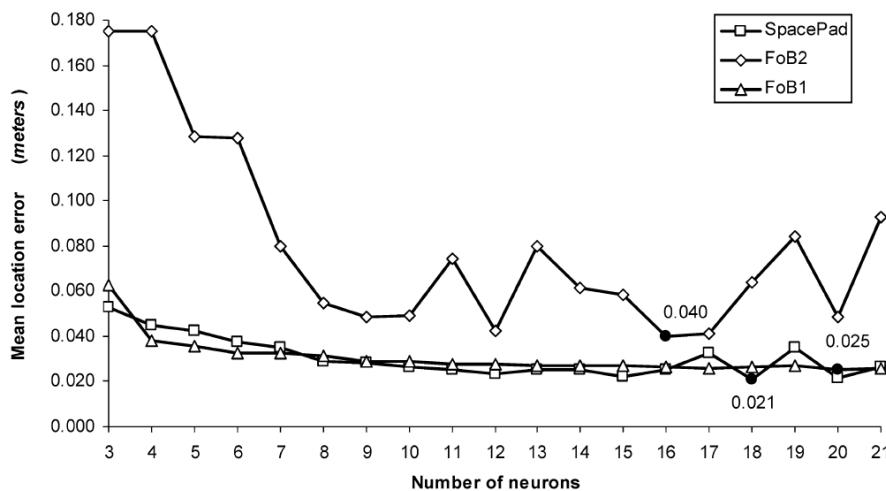


FIGURE 2.24 – L'erreur moyenne de position après la calibration en fonction du nombre de neurones, [KINDRATENKO et SHERMAN \[2005\]](#)

2.5 COMPARAISON DES TECHNIQUES DE CALIBRATION

Le tableau suivant 2.1, résume les différentes techniques de compensation d'erreurs statiques présentées dans la section précédente.

TABLEAU 2.1 – Résumé des différentes techniques de calibration, KINDRATENKO [2000], KINDRATENKO et SHERMAN [2005]

Méthode	Qualité de calibration en position	Qualité de calibration en orientation	Avantages et inconvénients
Interpolation tri linéaire	Moyenne	-	<ul style="list-style-type: none"> ☹ Limitations topologiques ☹ Discontinuité du gradient
Interpolation locale	Moyenne	Moyenne	<ul style="list-style-type: none"> ☹ Limitations topologiques ☹ Discontinuité du gradient
Méthode de Hardy HMQ	Bonne	Bonne	<ul style="list-style-type: none"> ☺ Mire non structurée peut être utilisée ☺ Représentation continue et régulière
Interpolation polynomiale	Bonne	Bonne	<ul style="list-style-type: none"> ☹ Augmentation occasionnelle de la valeur de l'erreur si l'erreur initiale est petite ☺ Mire non structurée peut être utilisée ☺ Représentation continue et régulière
Réseau de neurones	Très bonne	Moyenne	<ul style="list-style-type: none"> ☹ Paramètres de réseau non bien définis ☹ Le temps requis pour l'apprentissage ☺ Mire non structurée peut être utilisée ☺ Représentation continue et régulière

CONCLUSION

Nous avons présenté dans ce chapitre les différents protocoles et techniques utilisés afin de calibrer un système de tracking. Dans un premier temps, nous avons défini ce qu'est une calibration. Ensuite, nous avons souligné les difficultés rencontrées, d'une part pour caractériser ces systèmes et d'autre part, de proposer un protocole de caractérisation adéquat. Enfin, nous avons décrit les différentes techniques permettant de compenser les erreurs de tracking et procéder à la calibration de ces systèmes. Nous tenons à signaler que les travaux dont nous avons explicités dans cette section, sont ceux que nous avons choisis pour leurs similarités avec ceux de la réalité virtuelle, tant au niveau des systèmes employés que pour leurs applications. Suite à cette étude, nous pouvons en conclure, qu'il n'y a pas de technique générique ou universelle pour calibrer un système de tracking, et qu'il faut recourir à une caractérisation de chacun des systèmes avant même de décider quelle approche choisir. Cette caractérisation nous permet de mettre l'accent sur les avantages et les inconvénients du système de tracking. En fonction des résultats qui découlent de cette caractérisation, une personne, que l'on qualifiera d'expert, adoptera une ou plusieurs approches pour réaliser la meilleure calibration possible. Chaque approche a ses propres avantages et inconvénients. La totalité de ce chapitre souligne la difficulté d'utiliser les systèmes de tracking de manière fiable et surtout le choix d'une technique de calibration appropriée, puisque celle-ci est entièrement liée à la technologie utilisée, l'environnement dans lequel le système s'inscrit ainsi que ses conditions d'usage.

Néanmoins, la modélisation de type boîte, basée sur deux techniques de régression non linéaires, semble être une bonne solution pour palier au problème de calibration des systèmes de tracking. En effet, lorsque celui-ci est complexe et que son analyse à partir des équations physiques se révèle trop compliquée pour aboutir à un modèle de connaissance, une solution est d'établir un modèle boîte noire. Celui-ci est construit à partir un ensemble de données correspondant à des observations des sorties associées à un ensemble de vecteurs d'entrées connus. Dans le prochain chapitre, nous présenterons deux techniques de régression non linéaires.

Chapitre 3

Les techniques de régression non linéaire pour la calibration

Résumé

Ce troisième chapitre est consacré à la présentation des techniques de régression et plus particulièrement les techniques non linéaires. En effet, en l'absence d'un modèle analytique du système de tracking, une méthode de régression peut s'avérer utile pour la calibration : nous parlons alors d'une approche "boîte noire". C'est donc une approche de calibration boîte noire qui serait investiguée, sous la forme d'une étude comparative entre deux techniques de régression non linéaires : les réseaux de neurones et les machines à vecteurs de support.

Sommaire

Introduction	67
3.1 Les techniques de régression non linéaire	67
3.1.1 Les réseaux de neurones	67
3.1.1.1 Le neurone biologique	67
3.1.1.2 Les réseaux de neurones artificiels	68
3.1.1.2.1 Le neurone formel	69
3.1.1.2.2 Les réseaux de neurones artificiels	70
3.1.1.2.2.1 Les réseaux non bouclés	70
3.1.1.2.2.2 Les réseaux bouclés	71
3.1.1.2.3 L'apprentissage de réseaux de neurones artificiels	71
3.1.1.2.3.1 Apprentissage supervisé	71
3.1.1.2.3.2 Apprentissage non supervisé	72
3.1.1.3 Propriété fondamentale de la parcimonie des réseaux de neurones non bouclés à apprentissage supervisé	72
3.1.1.4 Le réseau perceptron multicouche MLP (Multi Layer Perceptron)	74
3.1.1.4.1 L'algorithme de rétropropagation du gradient de l'erreur	74
3.1.1.4.2 Configuration et optimisation des caractéristiques du réseau de neurones	77

3.1.1.4.2.1	Une, deux ou plusieurs couches cachées ?	78
3.1.1.4.2.2	Nombre de neurones dans la couche cachée	78
3.1.1.4.2.3	Les méthodes d'apprentissage :	79
3.1.2	Régression à vecteurs de support	82
3.1.2.1	Formulation de la régression à vecteurs de support	82
3.1.2.2	Vue d'ensemble	88
3.1.2.2.1	Réglage des hyper paramètres	89
3.1.2.2.2	Démarche pour la mise en œuvre des modèles basés sur les machines à vecteurs de support	90
3.1.2.3	SVR et réseaux de neurones	91
Conclusion		92

INTRODUCTION

Cette section présente une synthèse théorique et pratique des techniques de régression non linéaire et d'approximateurs universels (approximation d'une fonction dans un espace à n dimensions). Nous discuterons des réseaux de neurones non bouclés et plus particulièrement du réseau à couches multiples MLP (Multi-Layer Perceptron) en association avec la technique d'apprentissage de rétropropagation du gradient de l'erreur appelé "back-propagation". Les machines à vecteurs supports (SVM) sont également discutées en détail pour leurs qualités de génération de fonction et d'approximateurs universels.

3.1 LES TECHNIQUES DE RÉGRESSION NON LINÉAIRE

La régression est une approche statistique qui cherche à déterminer et analyser la relation d'une variable par rapport à une ou plusieurs autres. L'intérêt principal de cette approche est d'établir une approximation de la fonction qui relie les données en sorties aux données observées en entrées. Nous prenons parti de nous concentrer dans cette section sur les techniques de régression non linéaire et plus précisément les réseaux de neurones et les machines à vecteurs supports.

3.1.1 Les réseaux de neurones

Avant de se lancer, nous allons décrire le fonctionnement d'un neurone en débutant par sa nature biologique et en terminant par la modélisation qui en est une pâle d'approximation et porte le nom de neurone formel. Cette théorie vient par la suite ouvrir la voie de réseaux neurones artificiels (RNA) à couches multiples à connexions directes tels que le réseau de type MLP. Nous aborderons le perceptron multicouche avec ses différentes méthodes de convergence, à titre d'exemple la méthode de Levenberg-Marquardt qui est attirante du point de vue de la vitesse de convergence. L'unique inconvénient de cette méthode est l'évaluation approximative de la matrice Hessienne pour des réseaux non linéaires. Nous discuterons également de la capacité d'approximation parcimonieuse des réseaux de neurones formels sans toutefois entrer dans les détails mathématiques. Une problématique importante se situe au niveau du dimensionnement de réseaux de neurones dont la théorie est inexistante jusqu'aujourd'hui. C'est un domaine de recherche fortement ouvert mais plusieurs techniques provenant de la littérature permettrait d'optimiser le dimensionnement d'un réseau de neurones selon le contexte du problème. Nous incluons une courte partie théorique qui servira à optimiser le dimensionnement du réseau de neurones lors de la phase d'expérimentation.

3.1.1.1 *Le neurone biologique*

Un neurone est une cellule biologiquement spécialisée dans le traitement de l'information. La figure 3.1 présente une telle cellule. Le neurone est composé d'un corps cellulaire ou encore soma et deux prolongements cellulaires plus au moins ramifiés : les dendrites et l'axone. Le soma a un noyau qui intègre l'information génétique de l'organisme

dont le neurone est issu. Un neurone reçoit des impulsions en provenance d'autres neurones au niveau des dendrites et transmet l'information (pôle récepteur) à son tour via son axone (pôle émetteur). L'extrémité de l'axone peut être plus au moins ramifiée en arborisation terminale. Chaque branche de cette dernière est constituée d'une terminaison autour de laquelle se trouvent les synapses [GHADDAR et collab. \[2004\]](#). Une synapse est une composante élémentaire et une unité fonctionnelle de communication entre deux neurones.

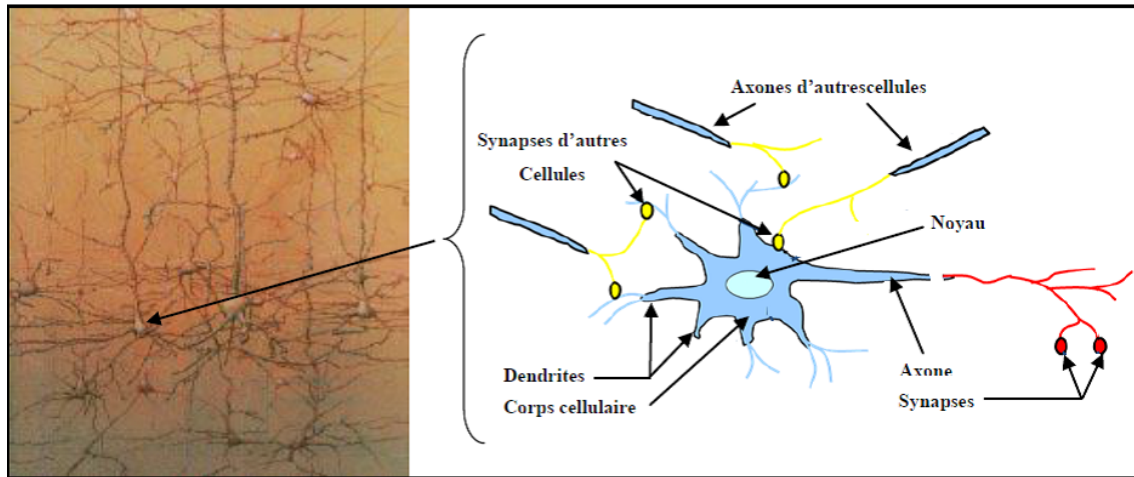


FIGURE 3.1 – Schéma de principe d'un neurone biologique

Le cortex cérébral humain est pellicule de neurones d'environ deux à trois millimètres d'épaisseur et d'une surface de mille deux cents millimètres. Le cortex contient donc environ 10^{11} neurones, soit l'équivalent du nombre d'étoiles dans la voie lactée. Les neurones sont interconnectés entre eux de façon bien dense et complexe. Nous estimons le nombre de liaisons entre un neurone et ses voisins entre mille et dix milles. Ce qui fait que le cortex cérébral intègre entre 10^{14} et 10^{15} de connexions.

3.1.1.2 Les réseaux de neurones artificiels

Les réseaux de neurones artificiels (RNA) sont des combinaisons fortement connectés de processeur de calcul. Chacun des processeurs de calcul est un neurone formel qui est, en soi, une formulation mathématique ou un modèle simplifié d'un neurone biologique.

Les réseaux de neurones artificiels disposent des capacités de mémorisation, de généralisation et d'une certaine forme d'apprentissage. Nous classons généralement les réseaux de neurones en deux familles : les réseaux non bouclés que nous appelons des réseaux non bouclés ou directs et les réseaux bouclés ou encore réseaux récurrents.

Dans ces deux familles, nous retrouvons des connexions totales ou partielles entre les couches. Le perceptron multicouche et le réseau RBF (Radial Basis Function) appartiennent à la famille des réseaux non bouclés. L'information transige seulement de l'entrée vers la sortie. Par contre les réseaux de [HOPFIELD \[1982\]](#) et ceux de [ELMAN \[1990\]](#) appartiennent à la famille des réseaux récurrents. Le réseau de [KASKI et collab. \[1998\]](#) est un autre exemple de réseaux récurrents.

Pour chacun de ces types de réseaux, il existe aussi des algorithmes d'apprentissage qui permettent de modifier les paramètres de poids et de biais.

Nous retrouvons beaucoup d'algorithmes d'apprentissage tels que l'algorithme du perceptron, la règle de Hebb, la règle de Widrow-Hoff (règle Delta) ainsi que des algo-

rithmes généralisés de Widrow-Hoff pour les réseaux multicouches (rétropropagation du gradient de l'erreur...).

Nous pouvons classer en deux grandes catégories les algorithmes d'apprentissage, soit l'apprentissage supervisé et l'apprentissage non supervisé; chacun ayant ses inconvénients et avantages. Pour être en mesure d'utiliser convenablement les réseaux artificiels de neurones, nous devons préparer les données qui se présentent en entrée et en sortie. Les phases principales qu'il ne faut pas considérer à la légère sont les phases de prétraitement et de post-traitement.

3.1.1.2.1) Le neurone formel

L'évolution des réseaux artificiels de neurones a été possible grâce à la volonté et aux recherches avant-gardistes de McCulloch et Pitts [1943]. Ils présentent la première modélisation d'un réseau de neurones. Selon les auteurs "Un neurone formel fait une somme pondérée des potentiels d'actions qui lui parviennent, ensuite s'active en fonction de la valeur de sommation pondérée. Si cette valeur dépasse un quelconque seuil, le neurone est activé et envoie une réponse dont la valeur est celle de son activation. Si le neurone n'est pas activé, il n'envoie rien".

Le neurone formel est un modèle mathématique très simplifié du neurone biologique. Le neurone formel est composé de plusieurs entrées dont chacune est multipliée par un poids. L'ensemble des résultats de chacune des entrées est présenté à un sommateur pour en former une combinaison linéaire. À la figure 3.2, nous schématisons un neurone formel avec ses entrées, ses poids ainsi que le module de sommation qui pondère les entrées.

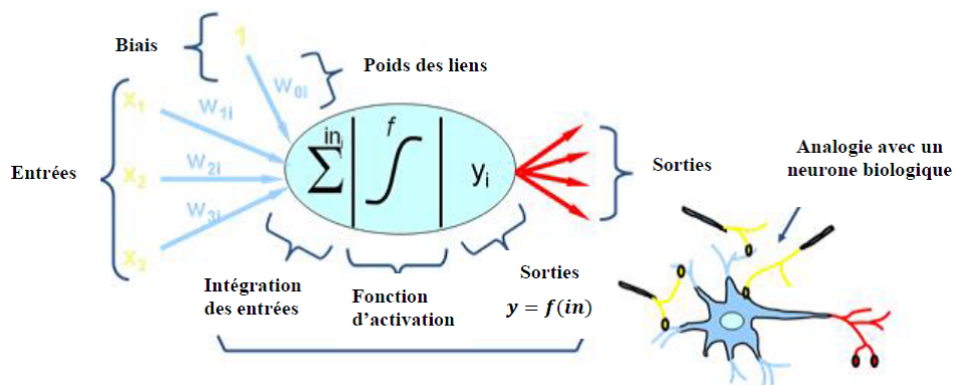


FIGURE 3.2 – Principe d'un neurone formel

La sortie $y_i(k)$ d'un neurone formel i est donnée par la relation suivante :

$$y_i(k) = f_i(o_i(k)) \quad (3.1)$$

Avec :
$$o_i(k) = \sum_{j=1}^n W_{ij}x_j(k) + \theta_i$$

Où θ_i est un biais

$f_i()$ est une fonction d'activation

$o_i(k)$ représente l'entrée totale,

$W_{ij} : 1, \dots, n$ sont les paramètres de pondération des connexions entre le neurone i et ses j entrées.

3.1.1.2.2) Les réseaux de neurones artificiels

Un réseau de neurone est un ensemble de neurones reliés entre eux par des connexions pondérées. Il se caractérise principalement par le type des unités utilisées et par sa topologie. Nous distinguons deux types de neurones particuliers : les neurones d'entrées recevant les données du monde extérieur et les neurones de sortie fournissant le résultat du traitement effectué. Les autres unités sont généralement qualifiées de "cachées". Comme nous le venons de voir, nous distinguons deux grandes catégories de réseaux de neurones artificiels : les réseaux non bouclés ou directs et les réseaux bouclés ou récurrents.

3.1.1.2.2.1 : Les réseaux non bouclés

Un réseau de neurones non bouclé est décrit par un groupe de neurones connectés entre eux, l'information transigeant des entrées vers les sorties sans retour en arrière. Si nous représentons le réseau par un graphe dont les nœuds sont les neurones et les arêtes sont les connexions entre ceux-ci, le graphe de ce type de réseau est acyclique. Si nous parcourons le réseau, à partir d'un neurone quelconque, en suivant les connexions, nous ne pouvons pas revenir au neurone de départ. Donc, la seule contrainte sur le graphe de connexion d'un réseau de neurones non bouclé est qu'il ne contienne pas de cycle. Une architecture très utilisée est celle du réseau multicouche. En effet, les neurones sont arrangés par couche (figure 3.3) et l'information dans le réseau, traverse toutes les couches et produit un résultat à sa sortie. Par conséquent, un réseau de neurones réalise une ou plusieurs fonctions algébriques de ses entrées et produit un résultat à sa sortie.

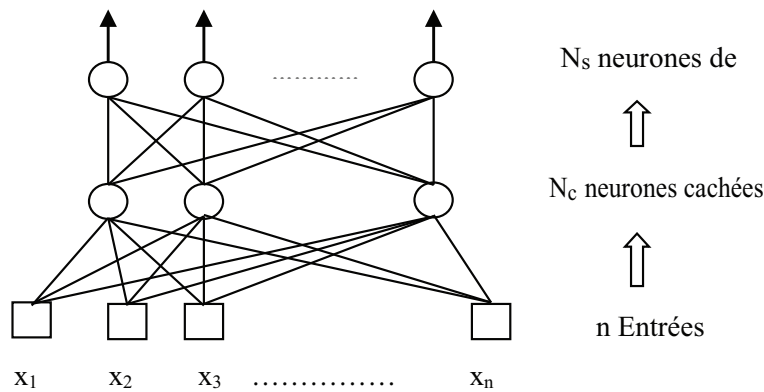


FIGURE 3.3 – Un réseau de neurones non bouclé à n entrées, une couche de N_c neurones cachés et N_s neurones de sortie

Considérons le cas, très important en pratique, des réseaux à une seule couche cachée à fonction d'activation sigmoïde et un neurone de sortie linéaire.

La figure 3.4 représente un exemple du réseau de neurones à $n + 1$ entrées, N_c neurones cachés, à fonction d'activation sigmoïde et un neurone de sortie linéaire.

La sortie du réseau a pour expression :

$$g(\mathbf{x}, \mathbf{w}) = \left[\sum_{i=1}^{N_c} w_{N_c+1,i} \text{th} \left(\sum_{j=1}^n w_{ij} x_j + w_{j0} \right) \right] + w_{N_c+1,0} = \left[\sum_{i=1}^{N_c} w_{N_c+1,i} \text{th} \left(\sum_{j=0}^n w_{ij} x_j \right) \right] + w_{N_c+1,0} \quad (3.2)$$

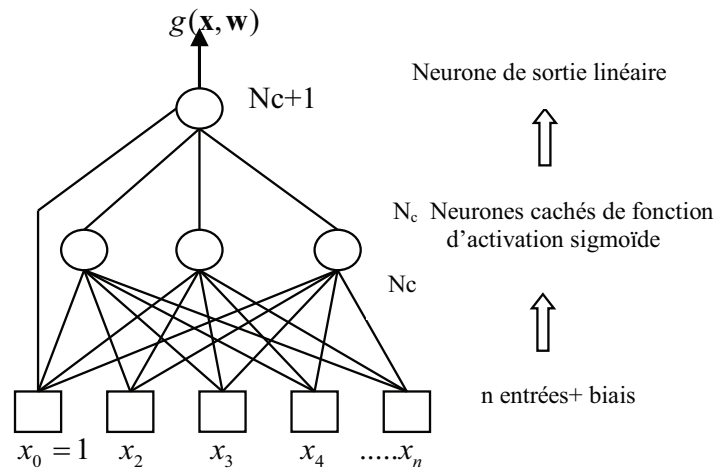


FIGURE 3.4 – Réseau de neurones à $n + 1$ entrées, N_c neurones cachés à fonction d'activation sigmoïde et un neurone de sortie linéaire

3.1.1.2.2.2 : Les réseaux bouclés

A l'inverse des réseaux de neurones non bouclés dont le graphe de connexion est acyclique, les réseaux bouclés peuvent emprunter une topologie de connexion quelconque, qui peut comprendre des boucles qui ramènent aux entrées la valeur d'une ou plusieurs sorties [DREYFUS et collab. \[2008\]](#). Lorsque nous parcourons le graphe en suivant le sens de connexion, il est possible de tomber sur un chemin qui revient à son point de départ : un tel chemin est désigné par le terme "cycle" (figure 3.5). La sortie de neurone peut être fonction d'elle-même, cela n'est trivialement concevable que si la notion de temps est explicitement prise en considération. Par conséquent, à chaque connexion d'un réseau de neurones bouclé est associé, outre un poids tel que pour les réseaux non bouclés, un retard multiple entier (éventuellement nul) de la période d'échantillonnage choisie. Une grandeur, à un instant donné, ne pouvant pas être fonction de sa propre valeur au même instant, tout cycle du graphe du réseau doit avoir un retard non nul.

3.1.1.2.3) L'apprentissage de réseaux de neurones artificiels

Nous appelons apprentissage de réseaux de neurones artificiels la procédure qui permet d'estimer les paramètres des neurones de réseau dans le but que ce dernier remplisse la tâche qui lui a été confiée. A la lumière de cette définition, nous distinguons deux types d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé.

3.1.1.2.3.1 : Apprentissage supervisé

Nous avons vu, dans la section précédente, qu'un réseau de neurones non bouclé peut représenter une relation algébrique entre ses entrées et ses sorties. Par conséquent, nous pouvons confier à ce réseau des tâches qui conduisent à réaliser une fonction algébrique non linéaire. Celle-ci peut être soit connue analytiquement et dans ce cas le réseau réalise une approximation de fonction, soit inconnue analytiquement, mais pour laquelle nous avons des mesures en nombre fini. Dans ce dernier cas le réseau réalise alors une tâche de régression.

C'est principalement pour ce dernier type d'application que sont mis en place les réseaux de neurones à apprentissage supervisé. La formulation du problème de l'apprentis-

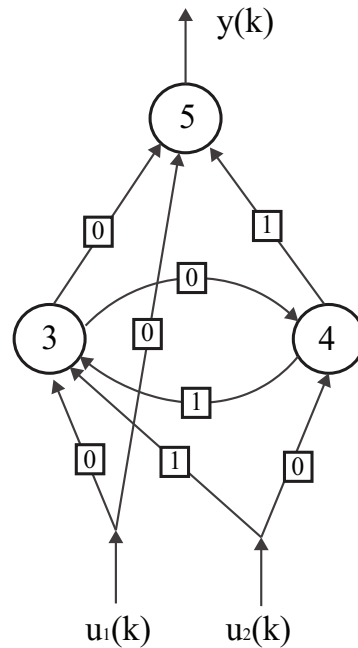


FIGURE 3.5 – Un réseau de neurones bouclé. Les nombres dans les carrés sont les retards associés à chaque connexion

sage supervisé est donc simple : nous disposons d'un nombre fini d'exemples d'une tâche à réaliser, sous forme de couples (entrée, sortie désirée), et nous souhaitons obtenir, d'une manière automatique, un système capable de trouver de façon relativement fiable la sortie correspondant à toute nouvelle entrée qui pourrait lui être présentée PASCAL [2003].

3.1.1.2.3.2 : Apprentissage non supervisé

Un réseau de neurones non bouclé peut être aussi utile pour la visualisation de données ou d'analyse de données. Nous disposons uniquement d'un nombre fini de données, constituées "d'entrées", que nous cherchons à les regrouper selon des critères de ressemblance inconnues à priori. Ce type de tâche est connue statistiquement sous le nom de techniques d'agrégation "clustering". Nous pouvons réaliser aussi une tâche assez voisine : à partir de données exprimées par des vecteurs de grande dimension, trouver une représentation de ces mêmes données dans un espace de dimension plus faible tout en gardant les "ressemblances" entre ces données.

3.1.1.3 Propriété fondamentale de la parcimonie des réseaux de neurones non bouclés à apprentissage supervisé

Les réseaux de neurones artificiels possèdent une propriété très importante qui est à l'origine de leur intérêt pratique dans des domaines très différents : ce sont des approximateurs universels parcimonieux. La propriété d'approximation peut être énoncée de la façon suivante :

"Toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire" HORNIK et collab. [1989] HORNIK [1991].

Cette propriété n'est pas propre aux réseaux de neurones : il existe d'autres familles

de fonctions paramétrées jouissant de cette propriété : nous citons à titre d'exemple les ondelettes, les fonctions radiales, les fonctions splines etc.

La particularité des réseaux de neurones est dans le caractère parcimonieux de l'approximation : à précision égale, les réseaux de neurones revendiquent moins de paramètres ajustables que les autres approximateurs universels. Plus particulièrement, le nombre de paramètres varie linéairement avec le nombre des variables de la fonction à approximer, bien qu'il vari d'une manière exponentielle pour les autres approximateurs **HORNIK et collab. [1994]**.

Qualitativement, la propriété de parcimonie peut se comprendre de la façon suivante **DREYFUS [1998]** : Quand l'approximation est une combinaison linéaire de fonctions élémentaires fixées (des monômes, ou des gaussiennes à centres et écarts-types fixes), nous ne pouvons ajuster que les coefficients de la combinaison. Par contre, quand l'approximation est une combinaison linéaire de fonctions non linéaires à paramètres ajustables (à titre d'exemple un perceptron multicouche ou MLP), nous ajustons à la fois les coefficients de la combinaison et la forme des fonctions que nous combinons. Par conséquent, dans un MLP les poids de la première couche précisent la forme de chacune des sigmoïdes réalisées par les neurones cachés, et les poids de la seconde couche fournissent une combinaison linéaire de ces fonctions. Nous concevons facilement que cette souplesse supplémentaire, conférée par le fait que nous ajustons la forme des fonctions que nous superposons, permet d'utiliser un plus petit nombre de fonctions élémentaires, en conséquence un plus petit nombre de paramètres ajustables.

Dans la pratique, nous n'utilisons pas les réseaux de neurones pour approximer des fonctions connues. Souvent, le problème qui se pose est le suivant : nous disposons d'un ensemble de mesures de variables d'un processus quelconque ainsi que du résultat du processus. Nous supposons qu'il existe une relation déterministe entre ces variables et ce résultat, et nous cherchons un modèle mathématique de cette relation valable où les mesures ont été réalisées. Il s'agit bien dans ce cas d'une modélisation "boîte noire".

En quoi l'approximation parcimonieuse peut-elle être utile dans la résolution de ce type de problème ? Ce que nous cherchons à obtenir à l'aide du modèle, c'est la vraie fonction, qui relie la grandeur y_p , que nous voulons modéliser, aux variables $\{x\}$ qui la déterminent. C'est-à-dire la fonction que nous obtiendrons en effectuant une infinité de mesures de y_p pour chaque valeur possible de $\{x\}$. En termes statistique, nous cherchons la fonction de régression de la grandeur à modéliser. Cette fonction est inconnue mais nous pouvons chercher une approximation à base des mesures disponibles. Cette approximation est réalisée en estimant les paramètres du réseau de neurones au cours de l'apprentissage. C'est ici que le caractère parcimonieux des réseaux de neurones est bénéfique. En effet, le nombre de mesures requis pour estimer les paramètres est d'autant plus important que le nombre de paramètres est important.

Par conséquent, pour modéliser, avec précision équivalente par le biais d'un réseau de neurones, il faut moins de données que pour la modéliser par le biais d'une régression multiple. Ainsi, un réseau de neurones permet, avec les données disponibles, de procéder à une approximation plus précise que la régression linéaire multiple.

Il en résulte qu'un réseau de neurones permet de mieux exploiter les mesures disponibles que des techniques de régression non linéaire conventionnelles. Rappelons que le nombre de paramètres (des mesures) varie d'une façon exponentielle pour les techniques de régression non linéaire conventionnelles bien qu'il varie d'une façon linéaire pour les réseaux de neurones. A la lumière de cette propriété remarquable, la technique de réseaux de neurones se présente comme une puissante technique de régression non linéaire.

3.1.1.4 Le réseau perceptron multicouche MLP (Multi Layer Perceptron)

Le réseau MLP est un modèle du réseau qui se compose d'une ou plusieurs couches cachées, dont les connexions sont directes et totales entre les couches. **WERBOS [1974]**, dans sa thèse de doctorat en 1974 explicita la théorie de rétropropagation du gradient de l'erreur. Cet algorithme fut popularisé en 1986 par Rumelhart qui l'utilise pour modifier les poids et les biais dans les réseaux de neurones artificiels multicouches. L'algorithme proposé modifie les paramètres et les biais en traversant le réseau de la couche de sortie vers la couche d'entrée en propageant l'erreur se trouvant à la couche de sortie vers chacune des couches antérieures. L'erreur de sortie étant calculée de la sortie réelle par rapport à la sortie désirée. L'apprentissage classique de ce type de réseau est effectué par une méthode de descente de gradient. Les poids sont initialisés de façon aléatoire et sont modifiés suivant une optique qui minimise une fonction de coût. Cette méthode présente cependant un défaut majeur, soit celui d'une probabilité importante de converger vers un minimum local qui faussera l'apprentissage. L'algorithme de rétropropagation du gradient de l'erreur "error back propagation" demeure l'algorithme le plus utilisé pour ce type de réseau. Il existe plusieurs variantes de l'algorithme de rétropropagation du gradient de l'erreur. La variante la plus intéressante étant l'algorithme de Levenberg-Marquardt appelé aussi "faster propagation".

3.1.1.4.1) L'algorithme de rétropropagation du gradient de l'erreur

Partons de la règle d'apprentissage de Widrow-Hoff et généralisons celle-ci pour un réseau à une couche cachée à connexions directes et totales entre les couches. A la couche cachée, la fonction d'activation est une fonction de type non linéaire. Habituellement, nous utilisons le sigmoïde. Cette fonction d'activation est souvent rencontrée dans les réseaux perceptron multicouche et sa dérivée est très simple. Pour les développements qui viendront, nous nous référons au réseau de la figure 3.6 qui représente un perceptron multicouche à une couche cachée.

Les développements mathématiques, pour un réseau à une couche cachée, seront plus simples à comprendre. Par convention, nous considérons ce réseau comme possédant deux couches de traitement, soit la couche cachée et la couche de sortie. Chaque neurone de la couche cachée réalise une somme pondérée de ses entrées pour former un scalaire que nous appelons "valeur d'activation de neurone". Nous le dénotons par net_j , où j désigne l'indice du neurone correspondant.

$$net_j = \sum_{i=1}^d w_{ij} x_i + w_{0i} = \sum_{i=0}^d w_{ij} x_i \equiv \hat{\mathbf{w}}_j^T \hat{\mathbf{x}} \quad (3.3)$$

La valeur d'activation est associée ensuite à une fonction d'activation f qui est une fonction facilement dérivable. Cette fonction est de la forme :

$$F(net_j) = \frac{1}{(1 + e^{-net_j})} = y_j \quad (3.4)$$

Nous remarquons que le résultat de cette fonction se situe entre 0 et 1. Lorsque la valeur d'activation d'un neurone est égale à zéro, la fonction d'activation donne la valeur 0.5; valeur qui sera transmise aux neurones de la couche de sortie. D'autre part, pour les données en entrée et la sortie désirée, dans les premiers débuts de l'apprentissage, il existe un écart entre la sortie réelle et la sortie désirée. Nous utilisons une fonction de coût quadratique qui agit sur tous les exemples de la base d'apprentissage.

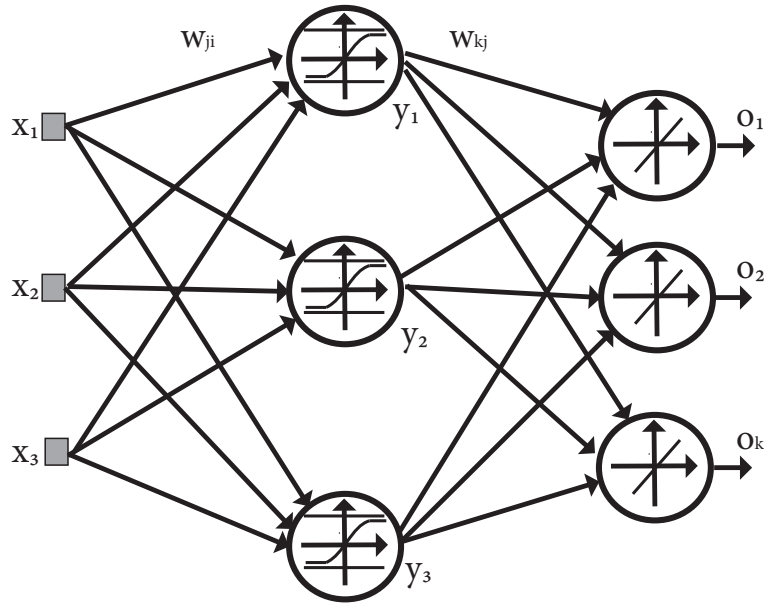


FIGURE 3.6 – Réseau MLP à une seule couche cachée

$$J(w) = \frac{1}{2} \sum_{i=1}^N (d_i - o_i)^2 \quad (3.5)$$

Où d_i est la valeur de la sortie désirée pour l'exemple i , et o_i est la valeur sortie du réseau pour l'exemple i et J est l'erreur quadratique entre la sortie désirée et la sortie retournée par le réseau. w est la matrice des poids.

Le concept fondamental utilisé par la rétropropagation du gradient de l'erreur est la variation de l'erreur quadratique en couche cachée par rapport à l'erreur quadratique de la couche de sortie. Afin d'ajuster les poids, nous utilisons la définition suivante, soit :

$$\Delta w = -\eta \frac{\partial J}{\partial w} \quad (3.6)$$

Nous pouvons exprimer également cette équation sous forme indiquée.

$$\Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}} \quad (3.7)$$

Où η dénote le pas d'adaptation ou d'apprentissage du réseau. La valeur est mise à jour avec la relation suivante :

$$w(m+1) = w(m) + \Delta w(m) \quad (3.8)$$

Où m est l'indice associée à la présentation d'une entrée particulière.

Evaluons maintenant ce que signifie la variation de l'erreur dans l'équation (3.8). Considérons les poids de la couche cachée vers la couche de sortie w_{kj} .

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} = -\delta_k \frac{\partial net_k}{\partial w_{kj}} \quad (3.9)$$

La sensibilité de l'unité k est exprimée comme suit :

$$\delta_k = -\frac{\partial J}{\partial net_k} \quad (3.10)$$

En présumant que la fonction d'activation est dérivable, nous différencions l'équation ci-dessus et nous prouvons que :

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial o_k} \frac{\partial o_k}{\partial net_k} = (d_k - o_k) f'(net_k) \quad (3.11)$$

La dernière dérivée partielle dans l'équation (3.11) est calculée en utilisant l'équation (3.4).

$$\frac{\partial net_k}{\partial w_{kj}} = y_j \quad (3.12)$$

La mise à jour des poids de la couche cachée vers la couche de sortie est décrite par l'équation suivante :

$$\Delta w_{kj} = \eta \delta_k y_j = \eta (d_k - o_k) f'(net_k) y_j \quad (3.13)$$

Dans notre cas, les neurones de sortie sont des fonctions linéaires telles que

$$f(net_k) = net_k \text{ et } f'(net_k) = 1 \quad (3.14)$$

alors que l'équation (3.13) se ramène à la règle Least Mean Square ou règle de Widrow-Hoff.

La règle d'apprentissage des entrées vers la couche cachée est un peu plus délicate. De la relation (3.5), en utilisant encore une fois la règle en chaîne pour les dérivées partielles, nous déterminons la variation de l'erreur par rapport aux poids w_{ji} .

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial y_j} \frac{\partial y_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} \quad (3.15)$$

Développons maintenant la première dérivée :

$$\frac{\partial J}{\partial y_j} = \frac{\partial}{\partial y_j} \left[\frac{1}{2} \sum_{k=1}^c (d_k - o_k)^2 \right] = - \sum_{k=1}^c (d_k - o_k) \frac{\partial o_k}{\partial y_j} \quad (3.16)$$

La variation de la sortie par rapport à l'entrée de la couche cachée s'écrit comme une nouvelle dérivée partielle :

$$\frac{\partial o_k}{\partial y_j} = \frac{\partial o_k}{\partial net_k} \frac{\partial net_k}{\partial y_j} = f'(net_k) w_{kj} \quad (3.17)$$

Substituons cette équation dans l'équation (3.13), nous obtenons la forme suivante :

$$\frac{\partial J}{\partial y_j} = - \sum_{k=1}^c (d_k - o_k) f'(net_k) w_{kj} \quad (3.18)$$

Par analogie avec l'équation (3.9), nous utilisons l'équation (3.16) pour définir la sensibilisé de l'erreur d'un neurone de la couche cachée :

$$\delta_j \equiv f'(net_k) \sum_{k=1}^c w_{kj} \delta_k \quad (3.19)$$

La relation (3.17) est le cœur de la solution qui concerne la connaissance de l'erreur de sortie se propageant vers les couches précédentes. La règle d'apprentissage pour les poids de l'entrée vers la couche cachée est la suivante :

$$\Delta w_{ji} = \eta \delta_j x_j = \eta \left[\sum_{k=1}^c w_{kj} \delta_k \right] f'(net_j) x_i \quad (3.20)$$

Les équations (3.16) et (3.20) constituent le cœur de l'algorithme de rétropropagation du gradient de l'erreur.

3.1.1.4.2) Configuration et optimisation des caractéristiques du réseau de neurones

Un aspect critique dans la conception des réseaux de neurones est le choix de la configuration appropriée du réseau pour une application donnée. Dans le cas des architectures multicouches, la configuration du réseau regroupe le nombre de couches, le nombre de neurones de nœuds par couche ainsi que le nombre de connections. En général, un réseau de neurones implémente une modélisation non linéaire.

$$u = G(x) \quad (3.21)$$

Le modèle est établi durant la phase d'apprentissage où le réseau de neurones apprend à associer correctement les entrées aux sorties. Etant donné un ensemble de paires d'exemples (x, u) , il y a probablement un nombre infini de configurations capables de relier les entrées x aux sorties u .

La question, qui se pose est la suivante : Quelle est la configuration la plus appropriée pour un problème donné ?.

Malheureusement, la réponse à cette question n'est pas toujours évidente. Plusieurs chercheurs confirment que la qualité de la solution trouvée dépend fortement de la configuration du réseau utilisée. Généralement, la taille du réseau affecte la complexité du réseau et le temps d'apprentissage. Elle affecte également les capacités de généralisation du réseau, la capacité de produire des résultats précis sur des données en dehors de l'ensemble d'apprentissage.

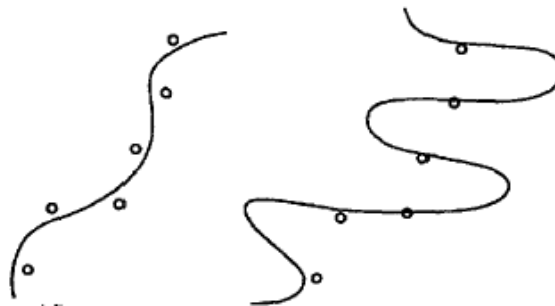


FIGURE 3.7 – Bon ajustement, Mauvais ajustement

Une analogie très illustrative entre l'apprentissage du réseau de neurones et l'ajustement de courbe (curve fitting) met l'accent sur l'importance de la taille du réseau de neurones. Il y a deux problèmes dans l'ajustement de courbe **BEBIS et GEORGIPOULOS [1994]** : détermination de l'ordre du polynôme et l'estimation des coefficients du polynôme. Par conséquent, étant donné un ensemble de points, nous choisissons l'ordre du polynôme et puis nous calculons ses coefficients. De cette manière, nous minimisons la somme des différences carrées entre les valeurs désirées et estimées. Une fois, les coefficients ont été calculés, nous évaluons tout point même les points qui n'ont pas fait partie de l'ensemble de données initial.

Si l'ordre de polynôme est très bas, l'approximateur obtenu n'est pas bon même pour les points faisant partie de l'ensemble de données initial. De l'autre côté, si l'ordre de polynôme choisi est très élevé, de très mauvaises valeurs peuvent être obtenus pour les points non inclus dans l'ensemble initial.

La figure 3.7 illustre ces concepts. Similairement, un réseau ayant une structure plus simple qu'il le faut ne peut pas fournir une bonne approximation même pour les points inclus dans l'ensemble d'apprentissage. Une structure plus complexe qu'il le faut peut entraîner un sur apprentissage de données : le réseau performe très bien sur l'ensemble d'apprentissage mais très mal pour des nouveaux points.

La prochaine section se propose de répondre à deux questions fondamentales dans la mise en œuvre d'un réseau de neurones non bouclé. Quel est le nombre de couches cachées à utiliser ? Quel est le nombre de neurones dans chaque couche cachée ?

3.1.1.4.2.1 : Une, deux ou plusieurs couches cachées ?

Une des premières décisions à prendre est de choisir le nombre de couches cachées nécessaires pour obtenir un bon modèle. En premier lieu, il convient de souligner que nous n'avons pas besoin de plus que deux couches cachées. Cette hypothèse est soutenue par les résultats théoriques et par de nombreuses simulations dans différents domaines d'ingénierie même s'il y avait des débats sur les réseaux avec trois couches cachées ou plus ayant des meilleures propriétés d'ajustement [HUANG et LIPPMANN \[1988\]](#). La vraie question est de savoir si une ou deux couches cachées doivent être utilisées. Une description claire de ce dilemme est abordée dans les travaux de [CHESTER \[1990\]](#) et [HAYASHI et collab. \[1990\]](#). Les deux papiers ont été publiés dans la même année mais ont été présentés dans deux conférences différentes. Le titre du premier est très explicite, "Pourquoi deux couches cachées sont meilleures qu'une seule?". Le deuxième, revendique que pour certains problèmes, un réseau de neurones à une seule couche cachée donne des performances meilleures. Il déclare "*Jamais essayer un modèle multicouche jusqu'à ce que vous avez essayé un modèle à une seule couche*".

Les deux architectures sont théoriquement capables d'approximer n'importe quelle fonction continue avec la précision désirée. Comme déjà mentionné, [CYBENKO \[1989\]](#), [FUNAHASHI \[1989\]](#), [HORNİK et collab. \[1989\]](#) ont prouvé indépendamment la propriété d'approximation pour un réseau de neurones à une seule couche cachée.

[KŮRKOVÁ \[1992\]](#) a donné une démonstration directe des capacités d'approximation universelle d'un réseau de neurones non bouclés à deux couches. Il a montré également comment estimer le nombre de neurones cachés en fonction de propriétés de la fonction à approximer et la précision souhaitée.

Il y a quelques indications que pour certains problèmes un réseau de petite taille avec deux couches cachées peut être utilisé quand un réseau à une seule couche nécessitait un nombre infini de nœuds. Par conséquent, il est difficile de dire quelle topologie est meilleure. Selon, [KECMAN \[2001\]](#), la règle, indiquant qu'il pourrait être utile de résoudre le problème en main en utilisant d'abord un réseau de neurones à une seule couche, semble appropriée.

3.1.1.4.2.2 : Nombre de neurones dans la couche cachée

Le nombre de neurones dans la couche cachée est un facteur important dans le dimensionnement d'un réseau de neurones par rapport aux capacités d'approximation du réseau. Le nombre de composantes d'entrée et le nombre de sorties sont généralement

déterminés par la nature de problème. Par conséquent, la puissance d'un réseau et sa capacité de généralisation sont déterminés par le nombre de neurones dans la couche cachée. Dans le cas de la régression non linéaire, la tâche principale est de modéliser la fonction qui relie les entrées aux sorties. En changeant le nombre de nœuds dans la couche cachée, deux situations extrêmes doivent être évitées : sous apprentissage (pas assez de neurones) et sur apprentissage de données ou modélisation du bruit (plusieurs neurones). En statistiques, ce problème est abordé sous la rubrique biais-variance dilemme. Un des outils statistiques utilisés pour surmonter ce problème est la validation croisée. L'idée principale est que le fait d'avoir des bons résultats sur la base d'apprentissage ne garantit pas une bonne capacité de généralisation. Par généralisation, nous désignons la capacité d'un réseau de neurones à fournir une solution correcte lorsque nous utilisons des données qui n'ont pas servi à l'apprentissage. L'approche la plus simple pour comparer différents réseaux est d'évaluer la fonction de coût avec des données qui sont indépendantes de celles utilisées dans l'apprentissage. Les réseaux de neurones sont entraînés en minimisant une fonction de coût appropriée définie à la base des données d'apprentissage. La performance de réseaux est ensuite mesurée en évaluant la fonction de coût avec un ensemble de validation indépendant et le réseau ayant l'erreur la plus réduite sur l'ensemble de validation est sélectionné. Cette approche est appelé "hold out". Etant donné que cette approche peut-elle même conduire à un sur-apprentissage, la performance du réseau doit être confirmée en mesurant ses performances sur un troisième ensemble indépendant appelé "ensemble de test" [BISHOP \[1995\]](#).

D'autres techniques ont été proposées pour déterminer le nombre de neurones cachés optimal : Principalement la technique constructive et la technique d'élagage. La première technique consiste à commencer avec un réseau de petite taille et de rajouter au fur et à mesure des nouveaux nœuds durant la phase de l'apprentissage [KWOK et YEUNG \[1997\]](#). En effet, les réseaux de petite taille peuvent tomber facilement dans un minimum local. L'ajout de neurones cachés permet de changer la forme de l'espace de poids et d'échapper donc au minimum local. Pour une meilleure compréhension, revenons sur le problème d'ajustement de courbe. Par analogie, l'approche constructive consiste à commencer avec un polynôme d'ordre réduit et à rajouter de termes d'ordre supérieur chaque fois que le polynôme choisi ne fournit pas un bon ajustement.

En revanche les techniques d'élagage, se proposent de trouver une solution rapide en commençant par un réseau de grande taille [REED \[1993\]](#) et le réduire au fur et à mesure. Par analogie avec l'ajustement de courbe, cette approche suppose que nous commençons par un polynôme d'ordre élevé et que nous éliminons graduellement les termes qui ne contribuent pas significativement à l'ajustement.

Une approche mixte qui intègre les deux méthodologies décrites précédemment a été proposée par [RIVALS et PERSONNAZ \[2003\]](#).

3.1.1.4.2.3 : Les méthodes d'apprentissage :

L'apprentissage supervisé consiste à donner au réseau des couples (entrée, sortie désirée). L'ensemble d'apprentissage est indispensable pour permettre à un algorithme de jouer sur les poids et fixer ceux-ci jusqu'à ce que les sorties du réseau convergent le plus possible des valeurs ciblées. Nous retrouvons deux protocoles d'apprentissage pour les réseaux multicouches : l'apprentissage non adaptatif ou par lot (batch-training) et l'apprentissage adaptatif ou en ligne (Online-training).

3.1.1.4.2.3.1 Apprentissage par lot Dans ce mode, les poids et les biais du réseau de neurones sont mis à jour après que l'ensemble complet de données d'apprentissage a été injecté au réseau. Les gradients calculés après chacun des exemples d'apprentissage sont additionnés pour calculer les changements de poids et biais.

3.1.1.4.2.3.2 Apprentissage en ligne L'apprentissage adaptatif est utile lorsque l'ensemble d'apprentissage est très élevé et que la mémoire devient prohibitive. Dans l'apprentissage en ligne, chacune des entrées est présentée une seule fois, ce qui peut sembler engendrer un apprentissage rapide mais nécessitera plus de données d'apprentissage.

3.1.1.4.2.3.3 Les méthodes d'optimisation Nous avons vu jusqu'à maintenant comment évaluer le gradient de la fonction de coût par rapport aux paramètres du modèle. Une fois que nous disposons de cette évaluation, nous effectuons une modification des paramètres, afin d'approcher d'un minimum de la fonction de coût. Nous examinons à présent quelques algorithmes de minimisation itérative des paramètres du modèle ou encore d'algorithmes d'optimisation.

La méthode du gradient simple consiste à modifier les paramètres par la formule suivante, à l'itération i de l'apprentissage :

$$w(i) = w(i - 1) - \mu_i \nabla J w(i - 1), \text{ avec } \mu_i > 0 \quad (3.22)$$

La direction de descente est donc simplement opposée à celle du gradient : c'est en effet la direction selon laquelle la fonction de coût diminue le plus rapidement. La quantité μ_i est appelée "pas du gradient" ou "pas d'apprentissage".

Cette méthode est simple, mais elle présente de nombreux inconvénients :

- Si le pas du gradient est trop petit, la décroissance du coût est très lente. Si le pas est trop grand, le coût peut augmenter ou osciller. Cette situation est illustrée sur la figure 3.8, qui représente les lignes de niveau de la fonction de coût (fonction de deux variables w_1 et w_2) et l'évolution du point représentatif du vecteur w au cours du déroulement de l'algorithme. Au voisinage d'un minimum de la fonction de coût, le gradient de cette dernière tend vers zéro : l'évolution du vecteur des coefficients devient donc très lente. Il en va de même si la fonction de coût présente des « plateaux » où sa pente est très faible ; ces plateaux peuvent être très éloignés d'un minimum, et, dans la pratique, il est impossible de savoir si une évolution très lente du gradient est due au fait que l'on est au voisinage d'un minimum, ou que l'on se trouve sur un plateau de la fonction de coût.
- Si la courbure de la surface de coût varie beaucoup, la direction du gradient peut être très différente de la direction qui mènerait vers le minimum ; c'est le cas si le minimum recherché se trouve dans une « vallée » longue et étroite (les courbes de niveau sont des ellipsoïdes allongés au voisinage du minimum), comme on le voit également sur la figure 3.8.

Pour porter remède au premier inconvénient, de très nombreuses heuristiques ont été proposées, avec des succès divers. Les méthodes de recherche unidimensionnelle (notamment celle qui est présentée dans les compléments théoriques et algorithmiques à la fin de ce chapitre), fondées sur des principes solides, sont recommandées.

Pour faire face aux deux autres problèmes, on utilise des méthodes du second ordre qui, au lieu de modifier les coefficients uniquement en fonction du gradient de la fonction

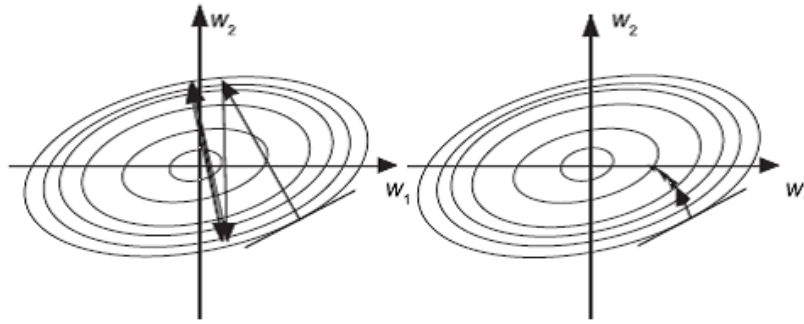


FIGURE 3.8 – Minimisation de la fonction coût par la méthode de gradient simple, DREYFUS et collab. [2008]

de coût, utilisent les dérivées secondes de cette dernière. Certaines de ces méthodes font également intervenir un paramètre μ susceptible d'être choisi à l'aide de méthodes de recherche unidimensionnelle.

Les grandes lignes des méthodes du second ordre les plus fréquemment utilisées, ainsi que les méthodes de recherche unidimensionnelle du pas, sont présentées brièvement dans ce qui suit.

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^2 \left(\frac{d^2 f}{dx^2} \right)_{x=x^*} + o(x^3) \quad (3.23)$$

Par conséquent, lorsque la variable x est au voisinage de x^* , on pourrait atteindre ce minimum en une seule itération si l'on connaissait la dérivée seconde de la fonction à son minimum : il suffirait pour cela de modifier la variable w de la quantité $\Delta x = \frac{\left(\frac{df}{dx} \right)}{\left(\frac{d^2 f}{dx^2} \right)_{x=x^*}}$

Le même raisonnement s'applique à une fonction de plusieurs variables, la dérivée seconde étant remplacée par la matrice hessienne $H(w)$ de la fonction à optimiser, de terme général $\frac{\partial^2 f}{\partial x_i \partial x_j}$ pour atteindre le minimum de la fonction de coût en une itération, il suffirait d'appliquer au vecteur des poids la modification suivante (sous réserve que la matrice hessienne soit inversible) :

$$\Delta x = -H(x^*) \nabla f(x) \quad (3.24)$$

Cette dernière formule n'est évidemment pas applicable en pratique, puisque le vecteur x^* n'est pas connu. Néanmoins, elle suggère plusieurs techniques qui mettent en oeuvre une approximation itérative de la matrice hessienne (ou de son inverse). Nous citons ici la méthode de Broyden-Fletcher-Goldfarb-Shanno (algorithme BFGS) BROYDEN [1970] et l'algorithme de Levenberg-Marquardt LEVENBERG [1944] MARQUARDT [1963]. Il faut noter que ces méthodes ne sont pas spécifiques aux réseaux de neurones : ce sont des méthodes très générales d'optimisation. Nous en trouverons des descriptions détaillées dans PRESS et collab. [1987]. Cet ouvrage présente également la technique du « gradient conjugué », qui utilise la matrice hessienne de manière implicite.

Pour l'apprentissage des réseaux de neurones, la fonction à optimiser $f(x)$ en fonction des variables x n'est autre que la fonction de coût $J(w)$, qu'il faut optimiser par rapport aux paramètres w du réseau. Le gradient de $J(w)$ est évalué par l'algorithme de rétropropagation décrit plus haut.

Que faire en pratique ?

En premier lieu, il ne faut pas utiliser la méthode du gradient simple et ses variantes, dont les temps de convergence (en nombre d'itérations) sont supérieurs de plusieurs

ordres de grandeur à ceux des méthodes du second ordre. L'utilisation de la méthode du gradient simple ne peut se justifier que dans le cas où l'on utilise de très gros réseaux (plusieurs milliers de paramètres). D'autre part, le choix entre les méthodes de BFGS et de Levenberg-Marquardt repose essentiellement sur des considérations relatives au temps de calcul et à la taille de la mémoire. La méthode de BFGS nécessite de choisir le moment où l'on passe du gradient simple à BFGS ; il n'y a pas, à cet effet, de règle fondée sur des arguments théoriques. Quelques tâtonnements sont parfois nécessaires pour trouver le « bon » nombre d'itérations (typiquement, une centaine), mais celui-ci n'est pas critique. La méthode de Levenberg-Marquardt ne présente pas cet inconvénient, mais elle devient lourde pour des « gros » réseaux (une centaine de paramètres), en raison de l'inversion de matrice nécessaire à chaque itération. Globalement, on a donc intérêt à choisir la méthode de Levenberg-Marquardt si le réseau est « petit », et celle de BFGS dans le cas contraire. La méthode du gradient conjugué peut également constituer une solution efficace au problème d'optimisation de la fonction de coût.

3.1.2 Régression à vecteurs de support

Les méthodes à vecteurs de support, Support Vector Machines (SVM) en anglais, ont été introduites par VAPNIK [1995] dans le cadre de la classification. Dans le cas de deux classes (ou catégories) linéairement séparables, une procédure de classification consiste à trouver un hyperplan tel que les données avec des étiquettes différentes soient de part et d'autre de cet hyperplan.

L'idée naturelle de Vapnik est de trouver l'hyperplan séparateur \hat{H} de telle sorte que la distance d entre les données et l'hyperplan soit maximale. Chercher \hat{H} maximisant d est un problème variationnel admettant une solution simple. Parmi tous les hyperplans séparateurs parallèles à \hat{H} , il existe nécessairement deux hyperplans, appelés marges, portant au moins une donnée. Les données portées par ces marges s'appellent vecteurs de support. L'un des aspects rendant les SVM populaires est que les vecteurs de support constituent en quelque sorte une représentation creuse des données (sparse représentation) puisque le problème de classification linéaire ne nécessite que la connaissance de ces vecteurs de support.

La méthode SVM développée pour la classification a donné naissance à une méthode similaire dans le cas de la régression. La régression par machines à vecteurs de support, support vector regression (SVR) en anglais, introduite par VLADIMIR N. VAPNIK [1997] et SMOLA et SCHÖLKOPF [1998], possède ainsi une propriété de représentation creuse des données. Toutefois, son intérêt principal est sans doute qu'il s'agit d'une méthode de régression robuste pour laquelle il existe des mises en œuvre efficaces. Les méthodes à vecteurs de support sont très bien comprises aujourd'hui et ont été traitées de manière extensive dans la littérature et les nombreux articles consacrés au sujet.

Les paragraphes suivants sont consacrés à une présentation plus détaillée de la méthode.

3.1.2.1 Formulation de la régression à vecteurs de support

Initialement développées pour résoudre les problèmes de classification, les machines à vecteurs de support peuvent être appliquées à la régression DRUCKER et collab. [1997] (pour un problème d'approximation de fonction). La régression, vu comme une tâche d'apprentissage supervisé, peut être formulée de la manière suivante : un ensemble de

données d'apprentissage l est fourni à la machine dans le but de déterminer la relation entre les entrées-sorties $f(x)$. L'ensemble de données d'apprentissage $D = \{[x(i), y(i)] \in \mathbb{R}^n \times \mathbb{R}, i = 1, \dots, l\}$ qui consiste à un ensemble de l paires $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l)$ où les entrées x sont des vecteurs à n dimensions $x \in \mathbb{R}^n$ et les sorties $y \in \mathbb{R}$ sont des valeurs continues. Nous introduisons les concepts clés de la régression par machines à vecteurs de support au fur et à mesure en commençant par une formulation de la régression linéaire :

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + b \quad (3.25)$$

Où w est le vecteur des poids sujet d'apprentissage et b est un scalaire.

Pour garantir la platitude de la fonction f , la norme des poids $\|\mathbf{w}\|$ est minimisée. Le problème revient à minimiser cette norme en garantissant que les erreurs soient inférieures à epsilon. Le problème peut donc s'écrire sous la forme d'un problème d'optimisation convexe :

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad (3.26)$$

Sous contrainte

$$|y_i - \mathbf{w}^T \mathbf{x}_i - b| \leq \epsilon, i = 1, \dots, l \quad (3.27)$$

Nous remarquons que dans cette formulation, l'objectif n'est pas de minimiser l'erreur d'apprentissage, comme dans le cas de réseaux de neurones, mais d'assurer la platitude de la fonction. La minimisation de l'erreur n'apparaît que sous la forme des contraintes. Autrement dit, aucune erreur n'est autorisée. La description donnée considère qu'une fonction linéaire f qui approxime toutes les données avec une précision epsilon. Dans la pratique, ce n'est pas toujours le cas. En présence du bruit important ou des valeurs aberrantes, il est primordial d'autoriser certaines erreurs.

Dan ce cas, nous utilisons la fonction de perte de Vpanik appelée également la fonction définie comme suit [KECMAN \[2005a\]](#) :

$$E(\mathbf{x}, y, f) = \begin{cases} 0 & \text{si } |y - f(\mathbf{x}, \mathbf{w})| \leq \epsilon \\ |y - f(\mathbf{x}, \mathbf{w})| - \epsilon, & \text{sinon} \end{cases} \quad (3.28)$$

Et comme

$$e(\mathbf{x}, y, f) = \max(0, |y - f(\mathbf{x}, \mathbf{w})| - \epsilon) \quad (3.29)$$

Par conséquent, la fonction de perte de Vapnik est égale à 0 si la différence entre l'estimation $f(\mathbf{x}_i, \mathbf{w})$ et la valeur mesurée y_i est inférieure à epsilon. La fonction de perte epsilon-insensible définit un ϵ -tube (figure 3.9). Si la valeur estimée est à l'intérieur de ce tube alors la perte est égale à zéro. Si la valeur estimée est en dehors du ϵ -tube la perte est égale à l'amplitude de la différence entre la valeur estimée et le rayon epsilon du tube.

La fonction de perte ϵ -insensible, qui peut être vue comme une variante de la fonction de Hubber, offre deux avantages : D'une part, la minimisation s'effectue facilement d'un de point de vue numérique et d'autre part, l'approximation à vecteurs de support est robuste vis-à-vis des données aberrantes.

Nous formulons l'algorithme de régression à vecteurs de support pour le cas linéaire d'abord. Ensuite, dans le but d'établir des modèles non linéaires, nous projetons les données en entrées dans un espace de très grande dimension appelé "espace de redescription" (feature space). Une solution qui semble se profiler : les noyaux qui sont un moyen efficace de travailler dans un espace de grande dimension à moindre coût [GEIST \[2013\]](#).

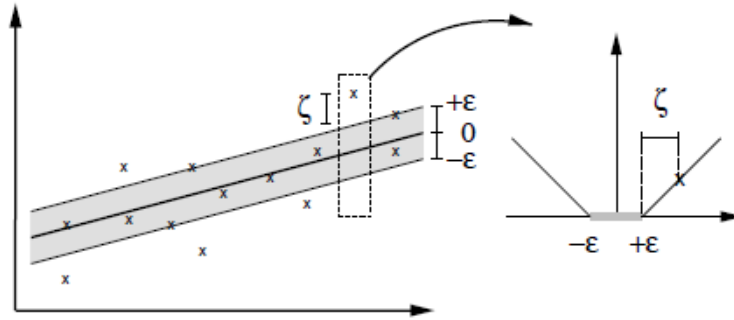


FIGURE 3.9 – La fonction de perte de Vapnik ϵ -sensible

Pour cela, nous mesurons l'erreur empirique R_{emp} en utilisant la fonction de perte de Vapnik ϵ -sensible donnée par la relation (3.30) et représenté dans la figure 3.9. Le risque empirique est exprimé par :

$$R_{emp}^{\epsilon}(\mathbf{w}, b) = \frac{1}{l} \sum_{i=1}^l |y_i - \mathbf{w}^T \mathbf{x}_i - b|_{\epsilon} \quad (3.30)$$

La figure 3.10 illustre deux fonctions linéaires d'approximation représentées en traits discontinus qui se situent à l'intérieur ϵ -tube et qui ont le même risque empirique que la fonction de régression $f(x, w)$ sur l'ensemble de données d'apprentissage.

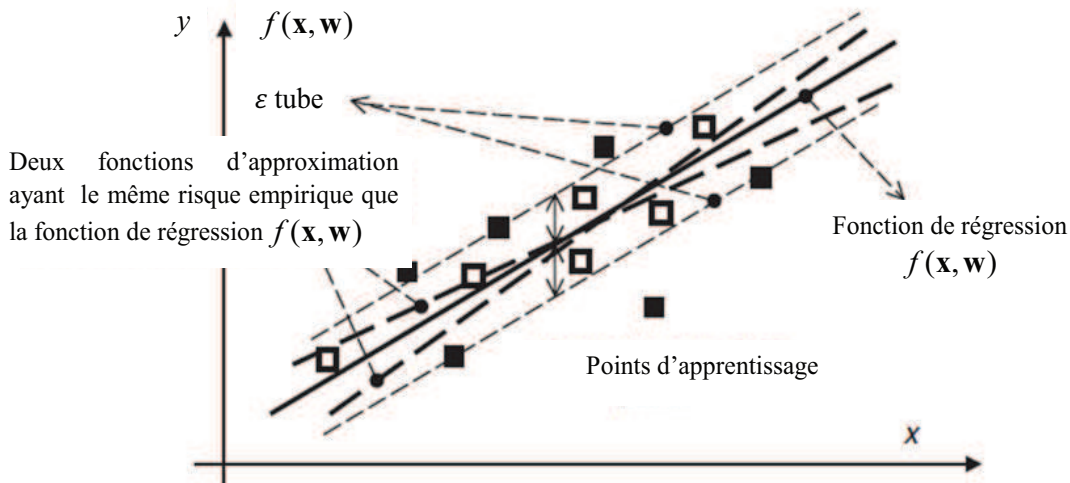


FIGURE 3.10 – Deux approximations linéaires à l'intérieur du tube ayant le même risque empirique R_{emp}^{ϵ} sur les données d'apprentissage que la fonction de régression. Les carrés remplis sont des vecteurs de support alors que les vides ne le sont pas.

Nous procédons maintenant à la minimisation à la fois du risque empirique R_{emp}^{ϵ} et $\|\mathbf{w}\|^2$. Par conséquent, nous construisons l'hyperplan de régression linéaire $f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x} + b$ en minimisant :

$$R = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l |y_i - f(\mathbf{x}_i, \mathbf{w})|_{\epsilon} \quad (3.31)$$

Où C représente le compromis entre régularisation et l'attache aux données.

Notons que cette expression ressemble à la régression rigide sauf que nous utilisons ici la fonction de perte ϵ -sensible de Vapnik au lieu de la fonction d'erreur quadratique.

De l'équation (3.28) et la figure 3.5, il en résulte pour toutes les données en dehors du tube de largeur epsilon que :

$$\begin{cases} |y - f(\mathbf{x}, \mathbf{w})| - \varepsilon = \xi & \text{si les données sont au dessus du tube} \\ |y - f(\mathbf{x}, \mathbf{w})| - \varepsilon = \xi^* & \text{si les données sont au dessous du tube} \end{cases} \quad (3.32)$$

Ainsi, minimiser le risque R revient à minimiser le risque suivant :

$$R_{\mathbf{w}, \xi, \xi^*} = \frac{1}{2} (\|\mathbf{w}\|^2 + C(\sum_{i=1}^l \xi_i + \sum_{i=1}^l \xi_i^*)) \quad (3.33)$$

Sous les contraintes :

$$\begin{aligned} y_i - \mathbf{w}^T \mathbf{x}_i - b &\leq \varepsilon + \xi_i & i = 1, l \\ y_i + \mathbf{w}^T \mathbf{x}_i - b &\leq \varepsilon + \xi_i^* & i = 1, l \\ \xi_i &\geq 0, \xi_i^* \geq 0 \end{aligned} \quad (3.34)$$

Où ξ et ξ^* sont des variables auxiliaires représentées dans la figure 3.9 pour tenir compte des points au dessus et au dessous du tube de largeur epsilon.

Les multiplieurs α_i et α_i^* de Lagrange (que nous introduisons durant l'optimisation ci-dessous) auront des valeurs différentes de zéro pour les données d'apprentissage qui sont respectivement au dessus ou au dessous du tube. Puisque aucune donnée ne peut être à la fois des deux côtés du tube, α_i et α_i^* sont par conséquent différents de zéro. Dans le cas où les points sont logés à l'intérieur du tube, les deux multiplieurs seront égaux à zéro et $\alpha_i \alpha_i^* = 0$.

Le lagrangien du problème énoncé ci-dessus peut s'écrire :

$$\begin{aligned} L_p(\mathbf{w}, b, \xi_i, \xi_i^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) &= \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l (\xi_i + \xi_i^*) - \sum_{i=1}^l (\beta_i \xi_i + \xi_i^* \beta_i^*) \\ &\quad - \sum_{i=1}^n \alpha_i (\mathbf{w}^T \mathbf{x}_i + b - y_i + \xi + \xi_i) \\ &\quad - \sum_{i=1}^n \alpha_i^* (y_i - \mathbf{w}^T \mathbf{x}_i - b + \xi + \xi_i^*) \end{aligned} \quad (3.35)$$

avec $\alpha_i, \alpha_i^*, \beta_i, \beta_i^* \geq 0 \forall i \in \{1, \dots, l\}$.

Le lagrangien $L(w, b, \xi_i, \xi_i^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)$ doit être minimisé par rapports aux variables primales \mathbf{w}, b, ξ_i et ξ_i^* et maximisé par rapport aux multiplicateurs de Lagrange non négatifs $\alpha_i, \alpha_i^*, \beta_i$ et β_i^* . La solution est fournie par le point-selle $(\mathbf{w}_0, b_0, \xi_{i0}$ et $\xi_{i0}^*)$. Au point-selle, les dérivées partielles du Lagrangien par rapport aux variables primales s'annulent. Il en résulte que :

$$\frac{\partial L_p(\mathbf{w}_0, b_0, \xi_{i0}, \xi_{i0}^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)}{\partial \mathbf{w}} = \mathbf{w}_0 - \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{x}_i = 0 \quad (3.36)$$

$$\frac{\partial L_p(\mathbf{w}_0, b_0, \xi_{i0}, \xi_{i0}^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)}{\partial b} = \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \quad (3.37)$$

$$\frac{\partial L_p(\mathbf{w}_0, b_0, \xi_{i0}, \xi_{i0}^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \quad (3.38)$$

$$\frac{\partial L_p(\mathbf{w}_0, b_0, \xi_{i0}, \xi_{i0}^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*)}{\partial \xi_i^*} = C - \alpha_i^* - \beta_i^* = 0 \quad (3.39)$$

Les équations (3.38) et (3.39) permettent d'éliminer les variables $\beta_i = C - \alpha_i$ et $\beta_i^* = C - \alpha_i^*$. En réinjectant, ces résultats dans (3.35), nous obtenons le lagrangien dual qui doit être maximisé :

$$\begin{aligned} L_d(\alpha_i, \alpha_i^*) &= -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \mathbf{x}_i^T \mathbf{x}_j - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i \\ &= -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^l (\varepsilon - y_i) \alpha_i + \sum_{i=1}^l (\varepsilon + \alpha_i^*) y_i \end{aligned} \quad (3.40)$$

Sous les contraintes :

$$\sum_{i=1}^l \alpha_i^* = \sum_{i=1}^l \alpha_i \text{ ou } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \quad (3.41)$$

$$\begin{aligned} 0 \leq \alpha_i \leq C \quad i = 1, l \\ 0 \leq \alpha_i^* \leq C \quad i = 1, l \end{aligned} \quad (3.42)$$

Grâce aux conditions complémentaires de Karush-Kuhn -Tucker selon lesquelles les produits entre les variables duales et primales sont nuls à la solution :

$$\alpha_i (\mathbf{w}^T \mathbf{x}_i + b - y_i + \varepsilon + \xi_i) = 0 \quad (3.43)$$

$$\alpha_i^* (-\mathbf{w}^T \mathbf{x}_i - b + y_i + \varepsilon + \xi_i^*) = 0 \quad (3.44)$$

$$\beta_i \xi_i = (C - \alpha_i) \xi_i = 0 \quad (3.45)$$

$$\beta_i^* \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0 \quad (3.46)$$

Les relations (3.45) et (3.46) permettent de déduire que pour $0 < \alpha_i < C$, $\xi_i = 0$ et que pour $0 < \alpha_i^* < C$, $\xi_i^* = 0$. De même, pour $0 < \alpha_i, \alpha_i^* < C$ et en se basant sur les équations (3.43) et (3.44), nous obtenons :

$$\mathbf{w}^T \mathbf{x}_i + b - y_i + \varepsilon = 0 \quad (3.47)$$

$$-\mathbf{w}^T \mathbf{x}_i - b + y_i + \varepsilon + \xi_i^* = 0 \quad (3.48)$$

Par conséquent, pour tous les points satisfaisants $y - f(x) = +\varepsilon$, les variables duales α_i doivent être comprises entre 0 et C ou $0 < \alpha_i < C$ et pour ceux satisfaisant $y - f(x) = -\varepsilon$, α_i^* prennent les valeurs $0 < \alpha_i^* < C$. A l'issue de cette remarque, nous pouvons calculer la valeur de biais b :

$$\begin{aligned} b &= y_i - \mathbf{w}^T \mathbf{x}_i - \varepsilon \quad \text{pour } 0 < \alpha_i < C \\ b &= y_i - \mathbf{w}^T \mathbf{x}_i + \varepsilon \quad \text{pour } 0 < \alpha_i^* < C \end{aligned} \quad (3.49)$$

Nous pouvons remarquer que d'après les relations (3.45) et (3.46), les seuls points pouvant être en dehors du tube de sensibilité sont ceux pour lesquels les multiplicateurs de Lagrange correspondants respectent $\alpha_i = C$ ou $\alpha_i^* = C$. D'autre part, pour les points logés à l'intérieur du tube ($|y_i - f(\mathbf{x}_i)| < \varepsilon$), nous avons $\alpha_i, \alpha_i^* = 0$ pour pouvoir vérifier les conditions (3.43) et (3.44). Cela engendre la propriété de parcimonie des SVMs : seuls les points avec des α_i, α_i^* correspondants non nuls sont nécessaires pour implémenter le modèle.

Ces points sont appelés vecteurs de support ou (Support Vectors) et ne présentent dans la pratique qu'une petite fraction de données d'apprentissage.

Après avoir calculé les multiplieurs de Lagrange α_i α_i^* et en utilisant l'équation (3.36), nous obtenons le vecteur de poids optimal correspondant à l'hyperplan de régression :

$$w_0 = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{x}_i \quad (3.50)$$

Et le meilleur hyperplan de régression est donné par le modèle :

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}_0^T \mathbf{x} + b = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \mathbf{x}_i^T \mathbf{x} + b \quad (3.51)$$

Il est maintenant temps de montrer l'utilité des SVM dans la régression non linéaire. En effet, l'extension des SVMs au cas non linéaire est simple et repose sur la projection des données dans un espace de dimension supérieure dans lequel le problème devient linéaire.

L'exemple issu de **SMOLA et SCHÖLKOPF [2004]** présente l'implémentation d'une fonction quadratique par un SVM linéaire. Soit la projection Φ définie par :

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3, \Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \quad (3.52)$$

L'algorithme SVM antérieurement décrit appliqué aux données projetées dans \mathbb{R}^3 par Φ fournirait une fonction linéaire \mathbb{R}^3 correspondant à une fonction quadratique dans \mathbb{R}^2 .

Cette approche semble intéressante en petites dimensions. Néanmoins, pour cet exemple, elle peut devenir coûteuse en temps de calcul pour des non linéarités polynômiales ou des problèmes à grande dimension d'entrée. Pour pallier ces problèmes, une projection implicite par fonction noyau est utilisée. Le fait que les données d'apprentissage n'apparaissent que sous forme de produit scalaire dans l'expression du problème d'optimisation (3.40) et dans le modèle de régression (3.51) constitue un des avantages majeurs de SVM.

Il n'est pas nécessaire de connaître la projection explicitement $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^f$ mais seulement le résultat $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$ du produit scalaire entre les images des points dans l'espace étendu F appelé aussi "feature space". Le problème non linéaire peut être donc reformulé ainsi :

$$L_d(\alpha_i, \alpha_i^*) = -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^l (\varepsilon - y_i) \alpha_i + \sum_{i=1}^l (\varepsilon + \alpha_i^*) y_i \quad (3.53)$$

Sous les contraintes

$$\sum_{i=1}^l \alpha_i^* = \sum_{i=1}^l \alpha_i \text{ ou } \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \quad (3.54)$$

$$\begin{aligned} 0 &\leq \alpha_i \leq C & i = 1, l \\ 0 &\leq \alpha_i^* \leq C & i = 1, l \end{aligned} \quad (3.55)$$

De même le vecteur de poids est donné par :

$$w_0 = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(\mathbf{x}_i) \quad (3.56)$$

et le modèle par :

$$f(\mathbf{x}, \mathbf{w}) = \mathbf{w}_0^T \Phi(\mathbf{x}) + b = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}) + b = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b \quad (3.57)$$

La différence centrale avec le cas linéaire est que le vecteur de poids w n'est plus exprimé explicitement dans \mathcal{R}^p mais implicitement dans F . De même, l'algorithme cherchera la fonction la plus plate dans l'espace étendu F et non plus dans l'espace d'entrée \mathcal{R}^p . La fonction k est appelée "fonction noyau" ou "noyau". Dans la pratique, seule cette fonction est connue ; la projection Φ ne l'est pas.

Les noyaux les plus souvent utilisés pour les SVMs sont les noyaux polynomiaux, sigmoïdaux et gaussiens radiale définis par :

- Linéaire : $k(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i)$
- Polynomial : $k(\mathbf{x}, \mathbf{x}_i) = [(\mathbf{x}^T \mathbf{x}_i) + 1]^d$ d dénote l'ordre du polynôme.
- Sigmoïdal : $k(\mathbf{x}, \mathbf{x}_i) = \tanh [(\mathbf{x}^T \mathbf{x}_i) + b]$
- Gaussien : $k(\mathbf{x}, \mathbf{x}_i) = \exp(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2})$

Pour être admissible, une fonction noyau doit répondre aux conditions de Mercer qui se résument à vérifier que la fonction $k : \mathcal{R}^{p \times p} \rightarrow \mathcal{R}$ correspond bien à un produit scalaire définissant une norme dans un espace étendu F .

3.1.2.2 Vue d'ensemble

Avant de s'intéresser à la mise en œuvre et aux détails de l'implémentation des SVR, résumons, brièvement, les différentes propriétés identifiées jusqu'ici.

La figure 3.11 illustre les différentes étapes de la régression.

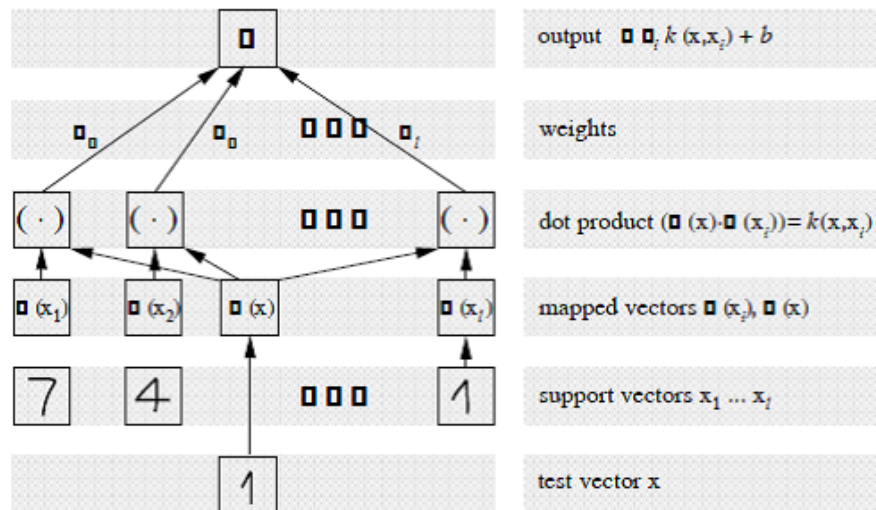


FIGURE 3.11 – Architecture de la méthode de régression basée sur les SVM

Les entrées sont projetées dans un espace de redescription en utilisant une projection implicite Φ . Le résultat du produit scalaire doit être les images des points dans l'espace étendu qui permet l'évaluation des fonctions noyaux $k(x, x_i)$. Enfin, les produits scalaires sont additionnés en utilisant les poids $v_i = \alpha_i - \alpha_i^*$. En rajoutant le biais b , nous obtenons le modèle de régression. Le processus décrit est similaire à celui utilisé par les réseaux de neurones, à la différence que dans le cas de vecteur de support les poids à la couche d'entrée sont un sous ensemble des données d'apprentissage.

3.1.2.2.1) Réglage des hyper paramètres

Dans la méthode de régression par machine à vecteurs de support, différents paramètres apparaissent : C est le compromis entre la complexité du modèle et l'erreur sur les données d'apprentissage, ε qui correspond au rayon du tube de sensibilité et les éventuels paramètres de la fonction noyau. Ces paramètres sont réglés en fonction d'une estimation de l'erreur de généralisation qui peut être évaluée sur un jeu indépendant de données de validation ou validation croisée [BISHOP \[1995\]](#). Cela conduit à réaliser l'apprentissage pour différentes valeurs et d'estimer leurs performances [CHUANG et collab. \[2002\]](#). Dans le cas d'estimation de l'erreur de généralisation par validation croisée, cette procédure s'avère coûteuse en temps de calcul. Une idée qui permet d'éviter le temps de calcul trop importants dans la recherche des hyper paramètres optimaux est donnée dans [SMOLA et SCHÖLKOPF \[2004\]](#). Si la nouvelle valeur C_{nouv} à tester est voisine de l'ancienne C_{anc} , le réajustement des multiplicateurs de Lagrange est utilisé pour l'initialisation du problème :

$$\alpha_{nouv} = \tau \alpha_{anc} \text{ et } b_{nouv} = \tau b_{anc} \quad (3.58)$$

En utilisant ainsi l'algorithme SMO (Sequential Minimal Optimization), un gain de 95% du temps de calcul peut être réalisé par rapport à un réapprentissage complet. D'autres approches permettent le réglage des hyper paramètres. Certaines sont basées sur des méthodes évolutionnistes [FRIEDRICHS et IGEL \[2005\]](#) ; les algorithmes génétiques [SAMANTA et collab. \[2003\]](#) et d'autres sur des méthodes hybrides [WANG et collab. \[2005\]](#) ; les recherches en grille (Search Grid) couplées avec l'exploitation des mesures de performance comme le critère de validation croisée ou critères sur les bornes [JUHOS et collab. \[2009\]](#).

La largeur du tube de sensibilité peut être effectué en considérant le niveau du bruit contenu dans les données. En effet, celui-ci détermine le niveau d'erreur qui ne sera pris en compte que par l'algorithme. Il existe une technique, appelée "v-SVR", permettant de construire un SVM qui ajuste automatiquement la largeur du tube. Elle consiste à ajouter un terme dans la formulation du problème d'optimisation pour tenter de minimiser ε . Le problème (3.31) est donc reformulé ainsi :

$$R_{w,\zeta,\zeta^*} = \frac{1}{2} (\|w\|^2 + C(\sum_{i=1}^l \zeta_i + \sum_{i=1}^l \zeta_i^* + N\nu\varepsilon)) \quad (3.59)$$

Sous contraintes

$$\begin{aligned} y_i - w^T x_i - b &\leq \varepsilon + \zeta_i & i = 1, l \\ y_i + w^T x_i - b &\leq \varepsilon + \zeta_i^* & i = 1, l \\ \zeta_i &\geq 0, \zeta_i^* &\geq 0 \end{aligned} \quad (3.60)$$

Avec $\nu > 0$ le problème dual est donnée par :

$$L_d^\nu(\alpha_i, \alpha_i^*) = -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^l (\alpha_i + \alpha_i^*) \quad (3.61)$$

Sous les contraintes suivantes :

$$\sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0, \sum_{i=1}^l (\alpha_i + \alpha_i^*) \leq C\nu N \quad \begin{aligned} 0 &\leq \alpha_i \leq C & i = 1, l \\ 0 &\leq \alpha_i^* \leq C & i = 1, l \end{aligned} \quad (3.62)$$

Le problème d'optimisation qui règle automatiquement epsilon est par conséquent similaire au problème d'origine avec un terme en moins dans le lagrangien dual et une

contrainte en plus. En outre, le paramètre ν choisi tel que $0 < \nu \leq 1$ peut être interprété à la fois comme le pourcentage d'erreurs en dehors du tube et le pourcentage des vecteurs supports à la fin de l'apprentissage comme démontré dans la référence [SCHÖLKOPF et collab. \[2000\]](#).

3.1.2.2.2) Démarche pour la mise en œuvre des modèles basés sur les machines à vecteurs de support

La démarche proposée s'articule principalement autour de trois étapes. La première étape consiste au pré-traitement des données c'est-à-dire :

- ✓ Nettoyer les données
 - Traiter les valeurs manquantes
 - Traiter les valeurs aberrantes
 - Traiter le bruit dans les données
- ✓ Transformer les données
 - Numériser les données
 - Normaliser les données

La deuxième étape concerne le développement du modèle c'est-à-dire :

- ✓ Choisir le type de noyau (RBF est un choix naturel).
- ✓ Déterminer les paramètres (C et γ pour les RBF) du noyau choisi. Utiliser la validation croisée et l'expérimentation pour déterminer les valeurs appropriées de ces paramètres.
- ✓ Si les résultats sont satisfaisants sinon changer le type ou les paramètres de noyau pour obtenir la précision voulue.

Finalement la troisième étape consiste à extraire et déployer le modèle.

Pour plus de clarté, une illustration de la démarche est représentée dans la figure 3.12. Une description succincte des étapes les plus importantes est présentée dans ce qui suit.

Normaliser les données (comme pour les réseaux de neurones) : La mise en échelle est une étape incontournable avant d'appliquer les machines à vecteurs de support. L'avantage majeur est d'éviter que les attributs ayant de grandes amplitudes dominent ceux ayant des faibles amplitudes. Un autre avantage est d'éviter les difficultés numériques pendant le calcul. Puisque les valeurs des noyaux dépendent des produits scalaires des vecteurs, par exemple le noyau linéaire et le noyau polynomial, les larges valeurs peuvent causer des problèmes numériques.

Choisir le noyau approprié : Bien qu'il existe principalement quatre types de noyaux, nous devons décider lequel utiliser dans un premier temps. [OLSON et DELEN \[2008\]](#) suggèrent que les noyaux gaussiens est un premier choix raisonnable. En effet, la non linéarité du noyau gaussien permet de projeter les données dans un espace de dimensions supérieures, à la différence du noyau linéaire ne peut pas gérer le cas où la relation à modéliser est non linéaire. Entre autre le noyau linéaire est un cas particulier de noyau gaussien. [KEERTHI et LIN \[2003\]](#) ont montré qu'un noyau linéaire avec un coefficient de pénalité C a les mêmes performances qu'un noyau gaussien avec les paramètres (C et γ). En outre, le noyau sigmoïdal se comporte comme un noyau gaussien pour certaines valeurs [LIN et LIN \[2003\]](#). La deuxième réponse est le nombre d'hyper paramètres qui influence la complexité de la sélection du modèle. Par exemple, le noyau polynomial a plus d'hyper paramètres que le noyau gaussien. Il en résulte que le noyau gaussien a moins de difficultés numériques.

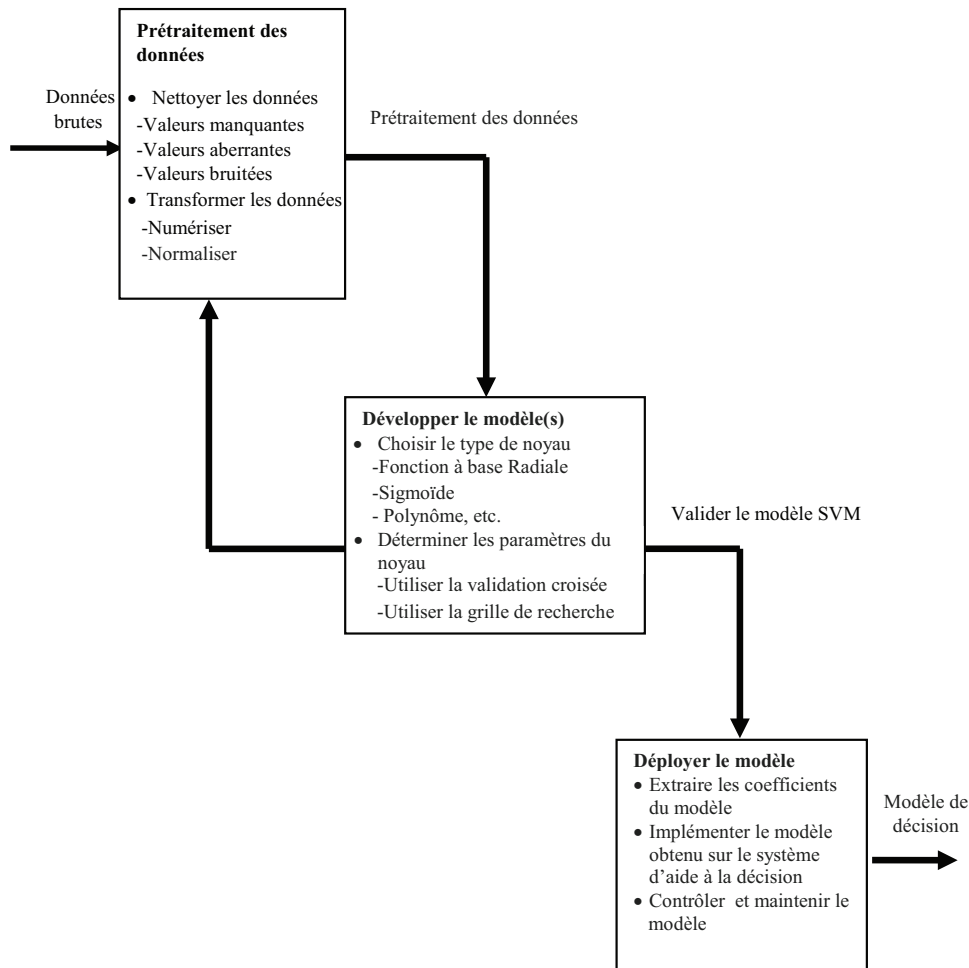


FIGURE 3.12 – Démarche de développement d'un modèle basée sur les SVMs

3.1.2.3 SVR et réseaux de neurones

Bien qu'il existe une certaine équivalence entre les SVMs et les réseaux de neurones régularisés [KECMAN \[2005b\]](#), les méthodes d'apprentissage restent différentes. En effet, l'apprentissage SVM apporte un certain nombre de réponses aux problèmes soulevés par l'apprentissage standard de réseaux de neurones.

- L'apprentissage correspond à un problème d'optimisation convexe ce qui entraîne l'unicité de la solution. En effet, la convexité est une propriété très importante des machines à vecteurs de support qui garantit l'unicité de minimum global contrairement au perceptron multicouche qui souffre de l'existence de plusieurs minima locaux.
- La fonction de sortie est douce (régularisation intrinsèque).
- Le problème de la dimension de données (Curse of dimensionality) est évité grâce à l'utilisation des noyaux permettant de résoudre le problème linéaire par une projection implicite dans un espace de dimension supérieure.
- L'inconvénient majeur est qu'il est difficile de choisir la structure du réseau de neurones (nombre de couches, nombre de neurones dans les couches cachées, procédure d'optimisation) alors que la taille de la machine à vecteurs de support est obtenue automatiquement par la sélection de vecteurs de support.
- Une bonne généralisation peut être obtenue à partir de peu de données.

CONCLUSION

Dans ce chapitre, nous avons mis en exergue deux techniques de régression non linéaire : les réseaux de neurones et la régression à vecteurs de support. Ces techniques permettent de faire une meilleure utilisation des mesures disponibles que les techniques de régression non linéaires conventionnelles. Elles apparaissent comme une puissante méthode de calibration. En effet ces deux techniques possèdent une propriété très importante qui est à l'origine de leur intérêt pratique dans des domaines très différents : ce sont des approximateurs universels parcimonieux. De plus, elles ont l'aptitude à estimer les données en sortie d'un système sans avoir connaissance à priori de son modèle analytique. En conclusion ces deux techniques semblent être une solution intéressante pour la calibration des systèmes de tracking n'ayant pas un modèle mathématique bien défini.

Chapitre 4

Mise en place d'un protocole expérimental pour la caractérisation du Scalable-SPIDAR

Résumé

Dans les chapitres précédents traitant la calibration des interfaces rencontrées en réalité virtuelle, nous avons déjà abordé le thème de caractérisation. Dans ce chapitre, nous proposons la mise en place d'un protocole de collecte de données permettant la caractérisation d'un système de tracking à échelle humaine : Le Scalable-SPIDAR. Pour ce protocole, nous pointons le plus de la réalité virtuelle dans la caractérisation des systèmes de tracking et démontrons que la multimodalité peut être une des solutions pour pallier certaines impossibilités techniques. Ce protocole répond notamment aux trois critères suivants : fiabilité, validité et commodité. Nous estimons louable l'idée de se servir des concepts et techniques de réalité virtuelle pour développer des protocoles de caractérisation des interfaces employées en réalité virtuelle.

Sommaire

Introduction	94
4.1 Plateforme expérimentale	94
4.2 Problèmes rencontrés dans le Scalable-SPIDAR	96
4.3 Caractérisation du Scalable-SPIDAR	98
4.3.1 L'initialisation multimodale du Scalable-SPIDAR	99
4.3.1.1 Paramètres multimodaux	100
4.3.1.2 Déroulement de l'initialisation du Scalable-SPIDAR	101
4.3.2 Protocole expérimental	102
4.3.2.1 Evaluation de l'ancien protocole	105
4.3.2.2 Protocole expérimental mis en place	106
4.3.2.3 Déroulement du nouveau protocole	109
4.3.2.4 Bilan de notre protocole de collecte de données	111
Conclusion	114

INTRODUCTION

Comme nous l'avons vu dans le premier chapitre, il existe une série d'interfaces matérielles utilisées en réalité virtuelle. Leur présence est fondamentale pour permettre à l'utilisateur d'interagir avec l'environnement virtuel. Néanmoins, comme nous l'avons vu dans le deuxième chapitre, ces interfaces et en particulier les systèmes de tracking, sont souvent loin de procurer des solutions fiables et satisfaisantes. En effet, elles sont souvent entachées d'erreurs dues à plusieurs contraintes notamment celles liées à la technologie utilisée et l'environnement dans lequel elles sont situées. En outre, il n'existe pas de technique générique ou universelle pour calibrer un système de tracking. Il faut donc recourir à une caractérisation de chacun des systèmes avant même de décider quelle approche adopter. Cette caractérisation nous permet de mettre l'accent sur les avantages et les inconvénients du système de tracking. L'objectif des recherches menées dans le domaine de calibration des interfaces de réalité virtuelle et plus particulièrement des systèmes de tracking est de proposer une procédure complète permettant la caractérisation et la compensation des erreurs de tracking. Actuellement, nous distinguons deux catégories de protocoles. Une première catégorie qui utilise des supports physiques ou virtuels pour l'acquisition des mesures de référence et une deuxième catégorie qui fait appel à un système de tracking secondaire pour la collecte des mesures de référence.

Dans ce chapitre, nous proposons un protocole expérimental original d'acquisition des données de position d'une interface à retour d'effort à grande échelle : le Scalable-SPIDAR. Le protocole proposé repose sur une stratégie de tracking hybride et permet, d'une part, de récolter d'une manière précise et guidée les mesures de position et d'autre part de caractériser les erreurs de position du Scalable-SPIDAR.

4.1 PLATEFORME EXPÉRIMENTALE

Comme nous l'avons mentionné, cette thèse a été accompagnée par la mise au point d'une plateforme expérimentale pour le tracking hybride. La plateforme proposée est basée sur un système de tracking optique infrarouge ARTtrack1 de Advanced Real-time Tracking. Ce dispositif, dit à "capture passive" est composé dans notre cas de deux caméras qui émettent des flashes infrarouges renvoyés par des sphères réfléchissantes appelées également marqueurs. Chaque caméra du système récupère une image de la position des marqueurs selon un point de vue différent. Ces images sont alors envoyées à un PC qui effectue une opération de triangulation afin de situer en trois dimensions chaque marqueur. Grâce à la détection des distances invariantes entre marqueurs 3D dans l'espace, reconnues lors de la triangulation, le système identifie des corps (systèmes de plusieurs marqueurs) ce qui conduit à connaître la position de certaines parties du corps de l'utilisateur comme les lunettes et donc la tête de l'utilisateur, ses bras ou encore ses mains. Ce système a l'avantage d'offrir une très bonne précision de l'ordre de dixième de millimètre et d'avoir un grand volume de travail (un cube d'environ trois mètres de côté par exemple pour le système ARTtrack1). Néanmoins, ce système souffre d'un inconvénient majeur : la perte de suivi. Les indices visuels du système peuvent être occultés partiellement ou totalement (corps humain, fumée...). Il est indispensable dans ce cas de basculer sur un autre système de tracking à grande échelle : le Scalable-SPIDAR. En effet, une interface à câbles à retour d'effort à grande échelle a été intégrée au sein de notre plateforme pour garantir la continuité des données et apporter un supplément de tangibilité et donc de réalité. En

effet, l'utilisation d'une interface à retour d'effort dans des applications de réalité virtuelle apporte un complément important à la restitution visuelle pour améliorer les qualités de perception, d'immersion et d'intelligibilité par l'utilisateur d'un environnement virtuel et pour lui donner la possibilité d'interagir physiquement avec les objets qui le constituent .

Le Scalable-SPIDAR est à la fois un système de tracking, il nous fournit la position et l'orientation de la main de l'utilisateur dans l'espace, mais aussi une interface à retour haptique. Il est composé d'une structure cubique sur laquelle des couples moteur/encodeur sont montés à chacune de ses extrémités. A chacun des moteurs est attaché un câble, les huit câbles convergent vers un effecteur qui peut être un anneau, une croix, ou une sphère. Au démarrage de ce système chacun des moteurs enroule son câble respectif afin d'exercer une tension minimale sur chacun d'eux. La somme vectorielle de toutes ces tensions fournit la tension exercée sur l'effecteur. Lorsque nous bougeons l'effecteur, la quantité de câble enroulée perçue par les huit encodeurs permet de dresser un système de huit équations à six 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 chacun des câbles et par conséquent la tension résultante sur l'effecteur engendrant ainsi un déplacement de celui-ci. Une des forces du Scalable-SPIDAR est le volume important couvert par l'espace de travail. En effet, comme nous l'avons vu dans le chapitre 1, **KUNZLER et RUNDE [2005]** ont démontré que le Scalable-SPIDAR offre un plus indéniable pour l'immersion et peut s'intégrer facilement au sein des environnements virtuels à échelle humaine. La sensation d'immersion est aussi meilleure que celle-ci offerte par les autres interfaces, car une fois devant l'écran, l'utilisateur distingue plus les câbles ce qui permet d'ignorer la présence du Scalable-SPIDAR. Ainsi, il est considéré comme transparent puisqu'il n'entrave pas les déplacements de l'utilisateur et n'induit pas une inertie qui pénalise ses mouvements. Entre autre, il offre un retour d'effort suffisamment puissant pour garantir une résistance forte (si l'objet dur) et d'une grande réactivité à la pression de l'utilisateur (passage d'une zone vide puis contact avec l'objet). Comme nous l'avons déjà mentionné, le Scalable-SPIDAR est doté d'un écran de grande taille qui offre une immersion de bonne qualité. Un tel couplage entre les informations visuelles et haptiques peut largement améliorer l'interaction de l'utilisateur avec les objets virtuels. Il s'avère très important pour garantir une bonne coordination œil-main lors de la manipulation des objets qui composent l'environnement virtuel. Toujours pour l'aspect immersif, la plateforme expérimentale permet aussi un rendu sonore grâce à un système d'enceintes. Celui-ci est prévu pour une spatialisation du son. Le rendu sonore est exploité pour renforcer les signaux qui sont annoncés par la modalité visuelle. La figure 4.1 décrit la plateforme expérimentale employée dans notre étude.

Néanmoins, le Scalable-SPIDAR souffre d'un problème majeur de répétabilité et de précision. En effet, les mesures obtenues sont souvent entachées d'erreurs. Ces erreurs sont dues essentiellement à la procédure d'initialisation adoptée et d'autre part aux incertitudes d'origine géométriques qui sont liés à la précision cinématique du modèle de l'interface : les imprécisions de conception ou de fabrication. Nous proposons dans la section suivante une étude détaillée de problèmes rencontrés dans le Scalable-SPIDAR afin de déceler les origines d'incertitudes sur son comportement.

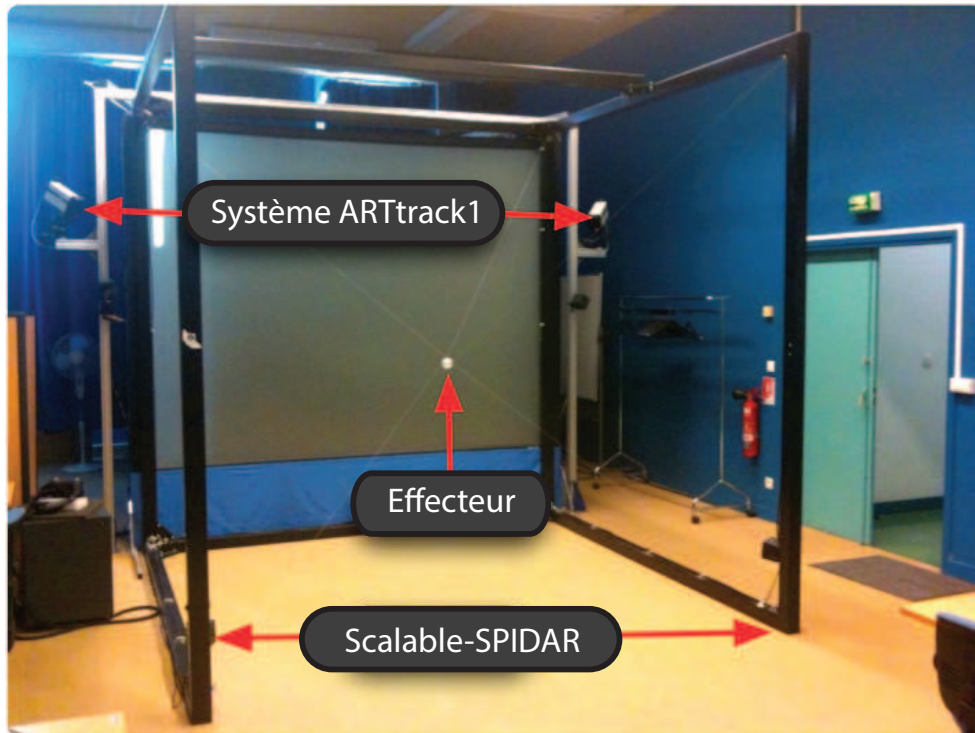


FIGURE 4.1 – Disposition de plateforme expérimentale

4.2 PROBLÈMES RENCONTRÉS DANS LE SCALABLE-SPIDAR

Comme nous l'avons mentionné précédemment, le Scalable-SPIDAR souffre d'une très mauvaise précision. Dans cette section, nous abordons les problèmes inhérents au Scalable-SPIDAR sous des angles différents afin de tendre vers des solutions pertinentes lors de la calibration.

Premièrement, le Scalable-SPIDAR est une interface qui requière une initialisation à chaque mise sous tension. Cette initialisation a une grande importance et permet de déterminer le référentiel utilisé par le Scalable-SPIDAR. Pour la réaliser, il faut placer l'effecteur au centre de sa structure avec la plus grande précision possible. En effet, une mauvaise initialisation provoque une détérioration des performances du système. Le Scalable-SPIDAR dispose d'une procédure d'initialisation qui consiste à déplacer l'effecteur de façon manuelle vers une position d'origine prédéfinie puis faire appel à un ensemble de fonctions implémentées dans le contrôleur du Scalable-SPIADR (High Definition Haptic Controller HDHC) pour positionner l'effecteur au centre de sa structure. Malheureusement, cette procédure n'est pas identique à chaque mise sous tension et par conséquent la précision de l'initialisation n'est plus assurée et ne peut être répétée de manière précise et faible. Cela nous met dans l'incapacité d'appliquer des traitements sur les mesures retournées par le Scalable-SPIDAR car tout traitement dépend du référentiel employé. Une procédure d'initialisation personnalisée est donc indispensable pour permettre à l'utilisateur de déplacer l'effecteur au bon endroit avec une bonne précision. Nous verrons dans la prochaine section la solution proposée pour obtenir une initialisation précise et répétable.

Les incertitudes d'origine géométrique nuisent, quant à elles, à la précision de la position et entâchent les mesures retournées par le Scalable-SPIDAR d'erreurs systématiques. Voici les problèmes de conception que nous avons identifiés :

- Les encodeurs se positionnent sur l'axe du moteur dont nous désirons calculer le nombre de tours. Avec cette méthode, le nombre de tours sont multipliés par le périmètre de câble pour obtenir la longueur du câble. Cette méthode pose problème puisque le rayon d'enroulement change pendant le déplacement de l'effecteur. En effet, avant la mise en route de Scalable-SPIDAR nous devons spécifier le diamètre de la poulie dans un fichier de configuration pour déterminer la position de l'effecteur. La méthode courante a pour effet de faire varier le diamètre de la poulie en fonction de la quantité de câble enroulé faussant ainsi la mesure des longueurs et par conséquent la position de l'effecteur.
- La poulie de sortie du moteur doit permettre l'enroulement de la longueur nécessaire du câble sur l'axe du moteur. Cette longueur doit permettre à l'utilisateur de déplacer sa main dans tout l'espace de travail. Le point le plus éloigné de l'espace par rapport à chaque moteur est son coin opposé. Par conséquent, la longueur extrême du câble est la diagonale du cube, qui est en pratique environ cinq mètres.
- La dimension de la poulie doit être suffisante pour fournir la longueur extrême soit 5 m. Cependant le diamètre des poulies est mal adapté à la longueur de chaque câble. Dans la situation, où une grande quantité de câble est enroulée, le diamètre de celle-ci varie de façon très remarquable. Cette variation est d'autant plus intense que leur diamètre est petit.

Pour illustrer ce propos, nous calculons les dimensions de la poulie suffisantes pour fournir la longueur $l_{totale} = 5m$. Certaines dimensions sont limitées par rapport à celles du moteur et les autres dimensions sont à calculer (figure 4.2). Le diamètre de la partie centrale est de $10mm$ et la distance entre les deux disques de poulie est de $9.6mm$, l'épaisseur du câble utilisé est de $0.3mm$ et la dimension de la poulie doit être suffisante pour fournir la longueur $l_{totale} = 5m$ pour le câble. Le nombre d'enroulements du câble dans un passage est : $k = \frac{9.6}{0.3} = 32$.

Or,

$$\begin{aligned}
 l_{total} &= \sum_{r=5(\Delta r=0.3)}^R 2\pi k r \\
 &= \sum_{r=5}^R 200.96 r \\
 &= 200.96 [5 + (5 + 0.3) + \dots + (5 + 0.3n)]
 \end{aligned} \tag{4.1}$$

$$\begin{aligned}
 \Rightarrow 5000 &= 200.96 [5 + (5 + 0.3) + \dots + (5 + 0.3n)] \\
 \Rightarrow 24.88 &= 5(n + 1) + \frac{0.3k(n+1)}{2} \\
 \Rightarrow 0 &= 0.15n^2 + 5.15n - 19.88 \\
 \Rightarrow n &= \frac{-5.15 \pm \sqrt{5.15^2 + 11.928}}{0.3} \\
 \Rightarrow n &= 3.50
 \end{aligned} \tag{4.2}$$

Où r est le rayon de la partie centrale et Δr est l'épaisseur du câble.

Par conséquent, il faut au moins 4 passages de câble sur la poulie. Cependant puisque le câble ne se rembobine pas régulièrement autour de la poulie, on considère 3 fois cette valeur, donc 12 passages. Ainsi le diamètre externe de la poulie est $10 + 2 \times 12 \times 0.3 = 17.2mm$.

Il est clair que le diamètre de la poulie calculé $17.2mm$ est nettement supérieur à celui utilisé dans la réalisation actuelle $14mm$. Ceci a pour conséquence de fausser le calcul de la quantité de fil enroulé.

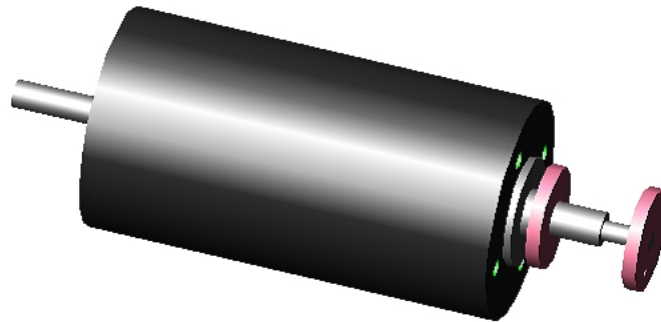


FIGURE 4.2 – Le moteur et la poulie sur son axe

- Les dimensions du Scalable-SPIDAR : tous les problèmes décelés antérieurement sont accentués par les grandes dimensions de notre Scalable-SPIDAR. Certains problèmes ont une légère influence lorsque le volume de travail a de petites dimensions deviennent non négligeables lorsque celui-ci a des dimensions plus importantes.
- Le Scalable-SPIDAR ne possède pas un guide qui conduit le câble tout au long de son enroulement ou déroulement. Ainsi, le câble est libre de superposer au fur et à mesure de l'enroulement ce qui peut entraîner le cisaillement du câble sur lui-même (figure 4.3). En outre, le câble peut facilement dérailler de la gorge de la poulie surtout lorsque l'effecteur se déplace d'une manière assez abrupte. En effet, si la poulie échoue à enrôler le câble rapidement, l'excès du câble à l'entrée se plie et forme une large courbure capable de ressortir le câble de la gorge de la poulie. Par conséquent la quantité enrôlée de câble ne reflète plus la distance entre l'extrémité du câble et l'effecteur. Là encore, la mesure de position est entachée d'erreurs.
- Un autre problème est que les câbles se détendent lorsque l'effecteur bouge vite. Cet inconvénient entraîne un problème de transparence du système. Ce délai de rembobinage des câbles accentue également l'imprécision du système. En effet, quand la main de l'utilisateur est dans l'espace libre, chacun des huit moteurs tire son câble pour le garder tendu. Afin de garantir une bonne transparence, cette tension est ajustée au minimum possible. Cependant, dans ce cas, les moteurs sont incapables de suivre les mouvements rapides de la main. Ceci est d'autant plus remarquable que la main a un champ plus grand et des mouvements plus rapides dans les environnements à échelle humaine tel que le Scalable-SPIDAR. Ainsi, le risque d'avoir des câbles non tendus est très probable. Lorsque, un câble se détend, le périphérique n'arrive plus à déterminer précisément la taille du câble et par conséquent la position de la main.

D'autre part, le Scalable-SPIDAR est doté d'une interface de programmation pour communiquer avec l'ordinateur. Néanmoins, cette interface nous permet seulement de récupérer certains paramètres du Scalable-SPIDAR mais en aucun cas nous ne pouvons agir sur le modèle permettant de calculer la position de l'effecteur. L'interface fournie est alors une boîte noire et cache ainsi le modèle cinématique interne de notre interface (figure 4.4). Nous verrons que cette situation peut diriger à la fois la conception de protocole d'acquisition de données mais aussi au choix de la technique de calibration.

4.3 CARACTÉRISATION DU SCALABLE-SPIDAR

L'idée centrale de notre approche est le développement d'une procédure complète pour la calibration du Scalable-SPIDAR. La procédure proposée est composée de trois

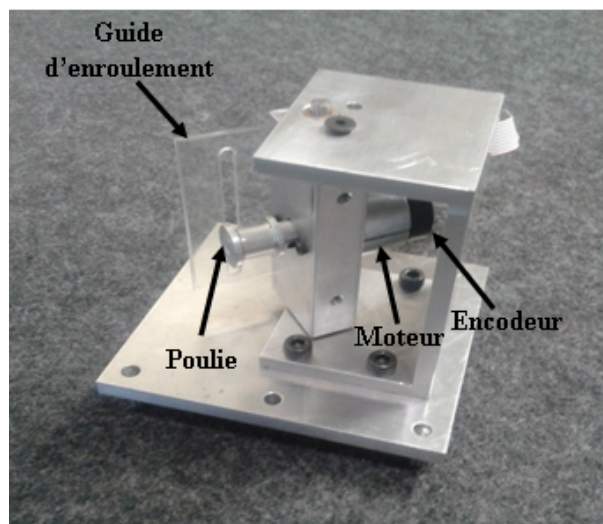


FIGURE 4.3 – Vue détaillée d'un moteur du Scalable-SPIDAR et du guide d'enroulement

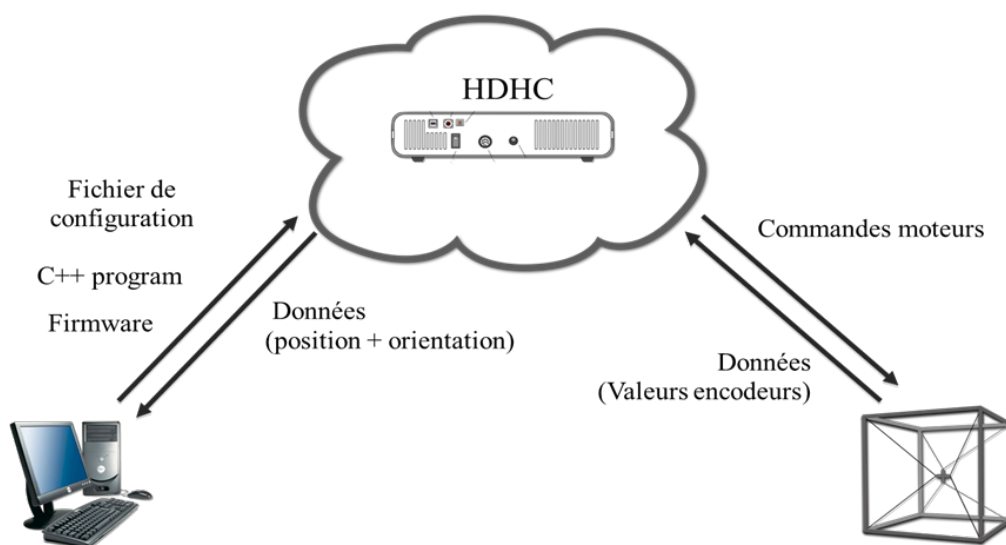


FIGURE 4.4 – Interface de communication avec le Scalable-SPIDAR (boîte noire)

grands modules. Le premier s'occupe de l'initialisation du Scalable-SPIDAR, le deuxième concerne la mise en place d'un protocole d'acquisition d'un protocole original d'acquisition des données permettant de caractériser les erreurs du Scalable-SPIDAR et évaluer l'influence des problèmes pointés dans la section précédente sur sa précision spatiale et enfin le troisième qui est dédié au développement d'une méthode de calibration basée sur les techniques de régression.

4.3.1 L'initialisation multimodale du Scalable-SPIDAR

Premier maillon de la procédure, l'initialisation du Scalable-SPIDAR. La procédure standard ne répond pas aux critères de précision et répétabilité décrits dans la section précédente. En effet, plus l'initialisation est précise plus les performances du Scalable-SPIDAR sont bonnes. Pour cela, nous proposons une procédure semi-automatique pour l'initialisation. Pour accomplir celle-ci avec la plus grande précision possible, nous utilisons le système de tracking optique : ARTtrack1. L'idée sous-jacente est de déterminer

les coordonnées de l'origine théorique du Scalable-SPIDAR dans le référentiel du système ARTtrack1 qui nous offre une précision de l'ordre du dixième de millimètre. Étant donné que nous connaissons la disposition géométrique de caméras infrarouges, l'origine du système de tracking optique ARTtrack1 ainsi que la disposition géométrique de la structure du Scalable-SPIDAR, nous avons pu déterminer les coordonnées de l'origine théorique du Scalable-SPIDAR. Une fois cette étape effectuée, nous avons cherché un moyen pour déterminer la position de l'effecteur de l'espace dans le référentiel ARTtrack1. Pour cela, une sphère réfléchissante est attachée à l'effecteur. De fait, le système de tracking ARTtrack1 est capable de déterminer la position de l'effecteur dans son référentiel. Le vecteur représente la distance qui sépare l'effecteur de l'origine théorique du Scalable-SPIDAR ainsi que la direction qu'il faut suivre pour converger à ce point.

4.3.1.1 Paramètres multimodaux

Afin de guider l'utilisateur à effectuer l'initialisation le plus justement possible et la répéter d'une manière précise, une assistance est indispensable. L'assistance consiste à offrir un retour multimodal (audio et haptique) à l'utilisateur afin de le guider pendant la phase d'initialisation.

Pour cela, il faut convertir les informations spatiales en paramètres multimodaux.

Nous définissons une fonction h , représentative de la distance d à laquelle nous nous trouvons de l'origine du Scalable-SPIDAR dans le référentiel du système ARTtrack1. En effet, plus la fonction $h(d)$ tend vers 1 plus nous sommes proches de l'origine. En revanche, si la fonction $h(d)$ tend vers 0 nous nous éloignons de l'origine. Nous définissons aussi $d_{min} = 1000mm$ la distance minimale à partir de laquelle la fonction commencer à varier, sinon $h(d)$ est égale à zéro.

$$h(d) = \begin{cases} \frac{d_{min}-d}{d_{min}} & \text{si } d \leq d_{min} \\ 0 & \text{si } d > d_{min} \end{cases} \quad (4.3)$$

$$d = \|\vec{v}\| \text{ avec } \vec{v} = [v_x, v_y, v_z]$$

Les assistances multimodales sont les suivantes :

La modalité auditive : Les enceintes émettent un signal sonore discret dont la fréquence change en s'approchant de l'origine du Scalable-SPIDAR (à la manière d'un radar de recul sur un véhicule). La distance d permet de déterminer la fréquence à laquelle les signaux sonores seront diffusés soit :

$$f(d) = h(d).1000 \text{ si } d \leq d_{min} \quad (4.4)$$

De fait, plus nous nous approchons de l'origine, plus la fréquence des bips augmente.

La modalité haptique : Un guide à modalité haptique est décrit principalement à l'aide de sa fonction d'effet. Celle-ci est une fonction qui retourne une force à transmettre à l'utilisateur en fonction d'un certain nombre de paramètres (position de l'effecteur, vitesse, constantes du modèle). La fonction d'effet comporte, quant à elle, deux composantes : composante de direction et composante intensité des forces.

En effet, à l'instant t , la fonction d'effet complète peut s'exprimer comme suit :

$$F(t) = F.\vec{e}_f \text{ avec } \vec{e}_f \text{ un vecteur unitaire } (\|\vec{e}_f\| = 1) \quad (4.5)$$

Nous définissons donc la fonction de direction des forces et le profil de force (fonction d'intensité des forces). La fonction de direction indique la direction de la force qui est

transmise à l'utilisateur. Le profil de forces est une fonction qui indique la puissance de la force transmise.

$$f_{dir}(x_1, x_2, \dots, x_n) = \|\vec{e}_f\| \text{ et } f_{pf}(x_1, x_2, \dots, x_n) = F \quad (4.6)$$

Avec x_1, x_2, \dots, x_n paramètres du système à l'instant t . Le profil de force est exprimé en fonction de paramètre de la distance d . Ce profil agit dans une zone restreinte autour de la zone d'intérêt, soit autour de l'origine de Scalable-SPIDAR. La fonction utilisée est continue, afin de ne pas brusquer l'utilisateur dans la tâche qu'il effectue.

$$s(x) = \frac{1}{1 + e^{-10(x-0.5)^2}} \text{ si } x \in [-0.5, 0.5] \quad (4.7)$$

La force retournée sera progressive et augmentera au fur et à mesure que nous nous approchons de l'origine.

Voyons maintenant, la fonction de direction. Un vecteur unitaire $\vec{\Delta}$, soit la normalisation du vecteur \vec{v} , indique la direction de la force qui sera rendue à l'utilisateur à la sortie du système. De fait, le vecteur-direction $\vec{\Delta}$ indique au système d'initialisation dans quelle direction il faut déplacer l'effecteur pour converger vers l'origine du Scalable-SPIDAR.

Enfin, nous obtenons la fonction d'effet complète :

$$\vec{F} = \vec{\Delta} \cdot s(d) \cdot F_{Max} \quad (4.8)$$

Où F_{Max} est la force maximale susceptible d'être appliquée sur l'effecteur.

La procédure d'initialisation proposée allie les deux modalités : auditive et haptique. Elle offre le double avantage d'être non intrusive et simple à implémenter.

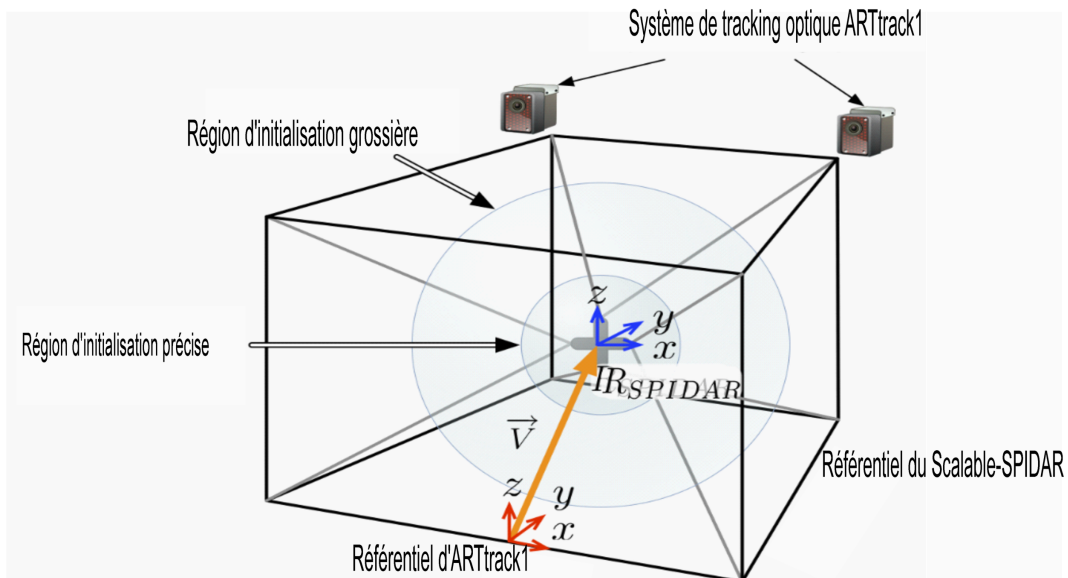


FIGURE 4.5 – Régions d'initialisation et cadre expérimental

4.3.1.2 Déroulement de l'initialisation du Scalable-SPIDAR

Une première initialisation, qualifiée de grossière, place l'effecteur du Scalable-SPIDAR relativement près du point d'initialisation. Avec cette première initialisation, nous arrivons à une précision de l'ordre de quelques centimètres. Cette première phase repose

sur l'utilisation d'une procédure d'initialisation proposée par le constructeur et implémentée dans le contrôleur du Scalable-SPIADR. Néanmoins avec cette procédure, nous ne pouvons pas obtenir une précision meilleure. En effet, le déplacement de l'effecteur, par l'application d'une force, ne peut s'effectuer de façon précise car le Scalable-SPIDAR ne possède pas une sensibilité suffisante. C'est donc pour cette raison que nous avons dû recourir à une deuxième phase, qualifiée d'initialisation précise (figure 4.6). Une fois l'initialisation grossière achevée, il nous faut nous rapprocher encore plus près du point d'initialisation. Afin d'y parvenir, nous avons utilisé une approche multimodale combinant la modalité auditive et haptique afin d'assister l'utilisateur à converger vers le point d'initialisation, dans une zone limitée autour du point d'initialisation. Cette approche permettrait de maximiser nos chances d'atteindre encore plus précisément et d'une manière répétable le point d'initialisation. Avec une telle méthode nous arrivons à une précision de l'ordre de 1 millimètre. Le processus d'initialisation est décrit par l'algorithme représenté par la figure 4.6.

Une fois l'initialisation effectuée, un autre problème reste à régler : le référentiel du tracking optique ARTTrack1 n'est pas le même que celui du Scalable-SPIDAR. Il est donc nécessaire de les faire correspondre. La position et l'orientation du tracking optique étant les plus précis et les plus avérés, il est plus logique de chercher à faire concorder le référentiel du Scalable-SPIDAR sur le référentiel du tracking optique ARTtrack1.

Pour cela, nous calculons le vecteur \vec{v} représentant la translation à effectuer pour faire coïncider le référentiel du Scalable-SPIDAR avec le référentiel du tracking optique ARTtrack1 (figure 4.6). Pour connaître ce vecteur, il suffit de calculer la différence entre la position de l'effecteur réel retournée par le tracking optique $P_{\text{effecteurARTtrack1}}$ avec celle retournée par le Scalable-SPIDAR $P_{\text{effecteurScalable-SPIDAR}}$. En d'autres termes :

$$\vec{v} = \overrightarrow{P_{\text{ARTtrack1}} P_{\text{Scalable-SPIDAR}}}$$

$$\vec{v} = \begin{pmatrix} P_{x_{\text{ARTtrack1}}} - P_{x_{\text{Scalable-SPIDAR}}} \\ P_{y_{\text{ARTtrack1}}} - P_{y_{\text{Scalable-SPIDAR}}} \\ P_{z_{\text{ARTtrack1}}} - P_{z_{\text{Scalable-SPIDAR}}} \end{pmatrix} \quad (4.9)$$

Une fois le vecteur \vec{v} calculé, il suffit de l'appliquer pour établir la correspondance entre les deux référentiels (figure 4.7).

4.3.2 Protocole expérimental

Comme nous l'avons vu dans le deuxième chapitre, les protocoles de collecte de données se distinguent par le type d'équipement utilisé. Il existe deux grandes catégories de protocoles. La première exploite les supports physiques ou virtuels pour recueillir les données de calibration. La deuxième s'appuie en revanche sur des systèmes de tracking secondaires ou encore une métrologie externe pour l'acquisition des mêmes données.

Pour notre protocole, notre choix s'est porté sur une stratégie de tracking hybride. A cela plusieurs raisons. La première est la présence d'un système de tracking non intrusif et très précis dans notre plateforme : le système de tracking optique infrarouge ARTtrack1. La deuxième provient des contraintes techniques puisqu'il est difficile de contraindre les mesures à saisir à des positions précises. Il est évident que la précision et la robustesse de la procédure complète de calibration du Scalable-SPIDAR est exclusivement dépendante de la qualité de calibration du système ARTtrack1. Une attention particulière est accordée à la calibration du système de tracking optique ARTtrack1. Dans cette section nous présentons la démarche suivie pour mener à bien la calibration. Deux types de calibration

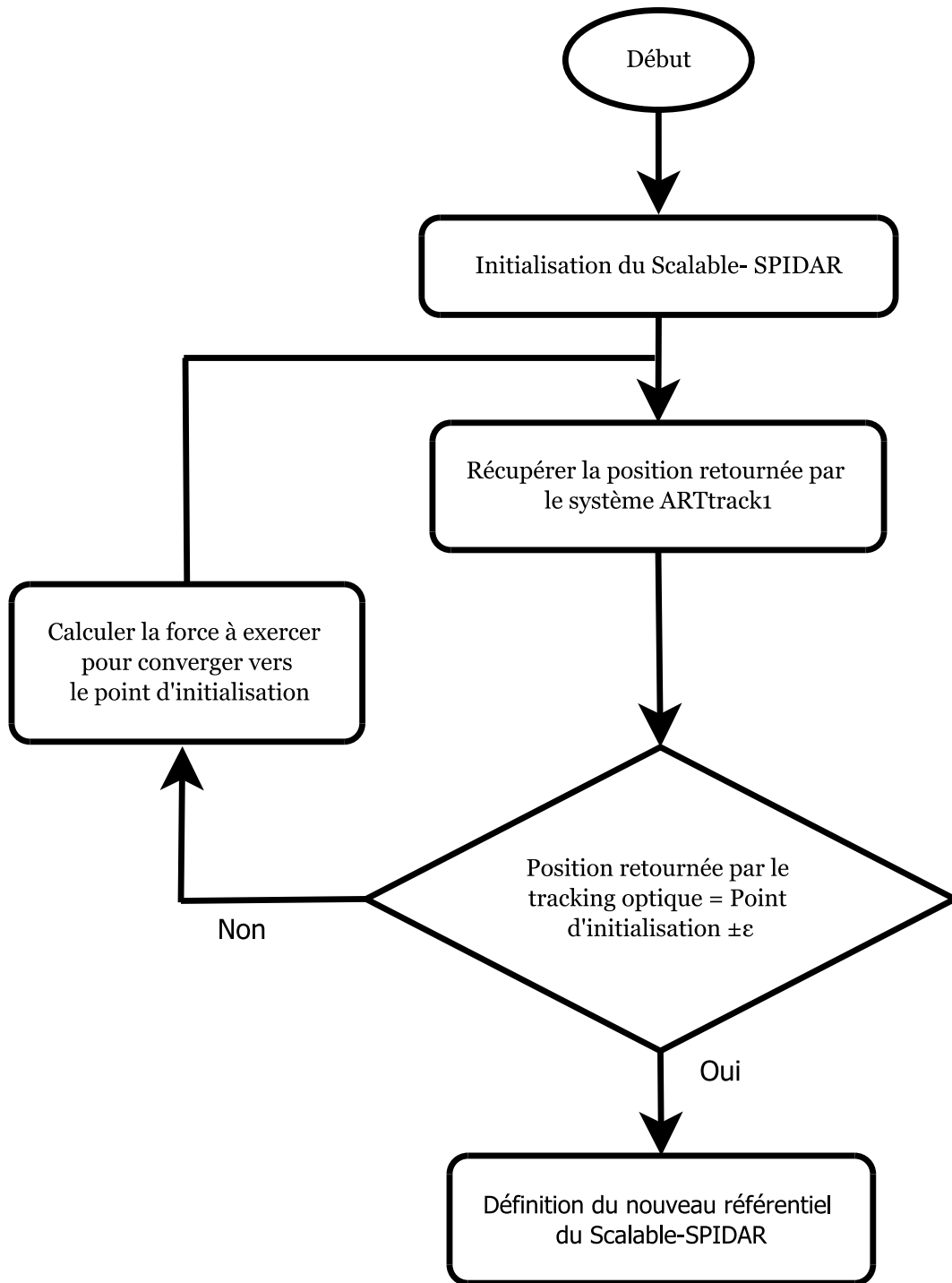


FIGURE 4.6 – Organigramme de la procédure d'initialisation du Scalable-SPIDAR

doivent être effectués : calibration de la pièce et la calibration du corps.

Pour effectuer la calibration de la pièce, l'angle de calibration est mis dans le champ de vision de deux caméras. Celui-ci permet de déterminer l'origine et les axes du référentiel du système de tracking ARTtrack1. Un bâton précalibré composé de deux sphères réfléchissantes pour créer un nuage virtuel dans l'environnement de mesure utilisé pour calculer les positions relatives des caméras infrarouges avec une grande précision. Pour plus de détails, nous renvoyons le lecteur intéressé vers le manuel d'utilisation [ART \[2003\]](#).

Pour plus de clarté, nous identifions l'origine et les axes du référentiel du système ARTtrack1 sur la figure 4.8.

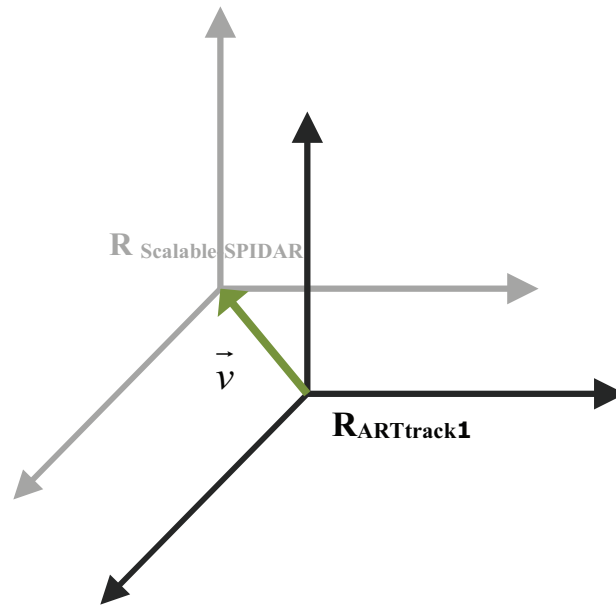


FIGURE 4.7 – Correspondance du référentiel du tracking optique ARTtrack1 avec celui du Scalable-SPIDAR

- Le centre du marqueur situé au point d'intersection de deux côtés de l'angle de calibration correspond à l'origine du référentiel du système ARTtrack1.
- Le grand côté de l'angle de calibration correspond à la direction +X
- Le petit côté de l'angle de calibration correspond à la direction +Y
- L'axe des z est ajouté de façon à créer un système de coordonnées orthogonal droit.

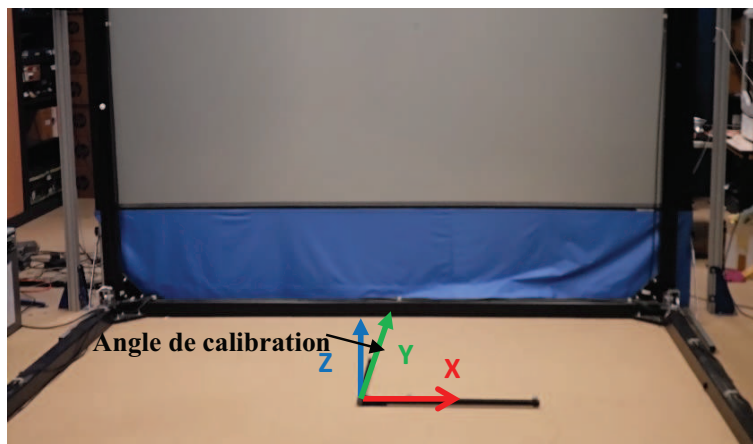


FIGURE 4.8 – Référentiel du système de tracking optique ARTtrack1

Durant la calibration du corps (body calibration), un système de coordonnées local est attribué à chaque corps rigide. Par conséquent, la position et l'orientation de la cible est déterminée par la transformation (\vec{x}, R) qui transforme un vecteur \vec{x} du système de coordonnées du corps au système de coordonnées de la pièce.

$$\vec{x}_{pièce} = R * \vec{x}_{corps} + \vec{s} \quad (4.10)$$

Ou \vec{x} désigne la position de l'origine du système de coordonnées du corps mesurée dans le référentiel de la pièce et R une matrice 3 * 3 décrivant la partie rotationnelle de la transformation. Une fois le système ARTtrack1 est calibré, la collecte de données à l'intérieur de l'espace de travail peut être lancée. Notre protocole se base en grande partie sur le protocole développé par BOUDOIN et collab. [2010] qui est lui-même inspiré de celui de GHAZISAEDY et collab. [1995]. Afin de mener une étude équilibrée, nous commençons tout d'abord par évaluer l'ancien protocole puis nous introduisons au fur et à mesure notre nouveau protocole.

4.3.2.1 Evaluation de l'ancien protocole

L'ancien protocole consiste à représenter une scène virtuelle composée de plusieurs petits cubes où chaque cube présente un sous espace de l'espace de travail. L'ensemble de ces cubes délimitait l'espace de travail du Scalable-SPIDAR (figure 4.9). Il permet également de guider l'utilisateur de manière à récolter un ensemble de positions qui sont réparties dans l'espace de travail. Ce protocole se déroule de la façon 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 entre en collision avec un cube, la position donnée par le Scalable-SPIDAR, la position donnée par le système ARTtrack1 ainsi que les valeurs provenant des différents encodeurs sont enregistrées.
3. Une fois ces données récupérées, le cube disparaît.

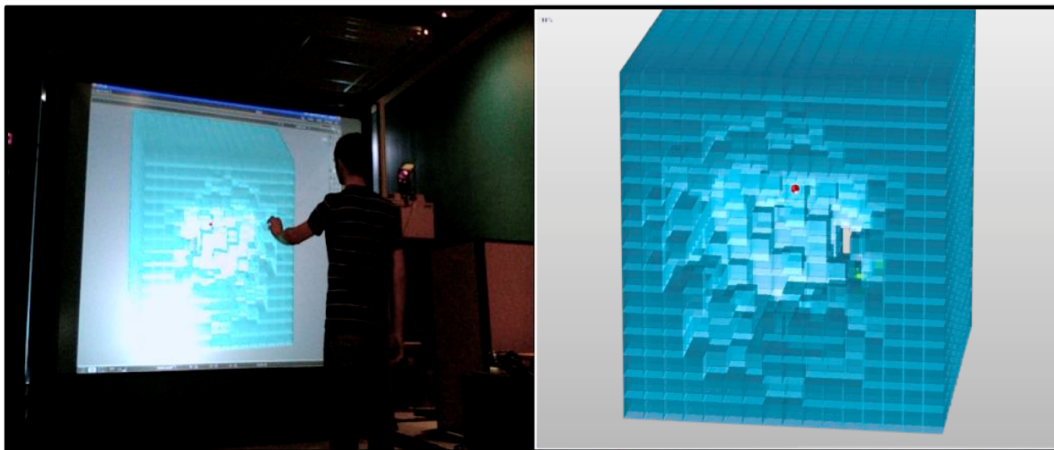


FIGURE 4.9 – Déroulement de l'ancien protocole. A gauche, un utilisateur est en train d'effectuer la saisie des positions. A droite, une illustration de ce que perçoit l'utilisateur

Bien que ce protocole semble probant en termes de validité et de commodité, il pêche au niveau de la fiabilité des mesures. En effet, nous avons remarqué qu'il y avait pas mal de redondance au niveau de données récoltées. Là, l'unicité des données n'est plus garantie.

De plus, comme nous l'avons constaté, aucune méthode de collecte de données n'est adoptée. La collecte étant aléatoire et non contrôlée et risque par conséquent de provoquer des déplacements brusques et abrupts de l'effecteur lors de la collecte de données. Toutefois, tel qu'il était conçu, le protocole suppose que la collecte des données se fait d'une manière quasi-statique. Ainsi, l'absence d'une méthode de collecte de données qui contraint les mesures à saisir à des positions précises nuit au caractère quasi-statique du

protocole et mène donc à des erreurs. Par ailleurs, comme nous l'avons déjà précisé, à chaque fois qu'une collision entre l'effecteur virtuel (représentée par une sphère rouge) et un cube est détectée, l'enregistrement à la fois des positions renvoyées par le Scalable-SPIDAR et par le système de tracking optique ARTtrack1 est lancé. Néanmoins, la technique de détection de collision employée se base sur la détection d'une intersection de type face à face. Ce type d'intersection est loin d'être précis et fait que le point collecté est souvent décalé d'une certaine distance du centre du cube. Là encore, la reproductibilité de l'expérience n'est plus assurée. La chaîne protocolaire souffre, quant à elle, de quelques limites qui empêchent de regrouper les données du Scalable-SPIDAR et du système ARTtrack1 au niveau d'un seul composant appelé collecteur de données. C'est donc pour des raisons d'efficacité et de fiabilité que nous avons décidé de revoir ce protocole et palier à ces défauts dans l'ambition de mettre en place un nouveau protocole plus précis et fiable.

4.3.2.2 Protocole expérimental mis en place

Nous présentons dans cette section le protocole qui a été implanté dans notre atelier de réalité virtuelle. Ce protocole doit permettre de caractériser les erreurs de position du Scalable-SPIDAR et mettre en exergue l'influence des problèmes de conception mécaniques sur sa précision. La figure 4.10 présente une version simplifiée de l'architecture du protocole proposé.

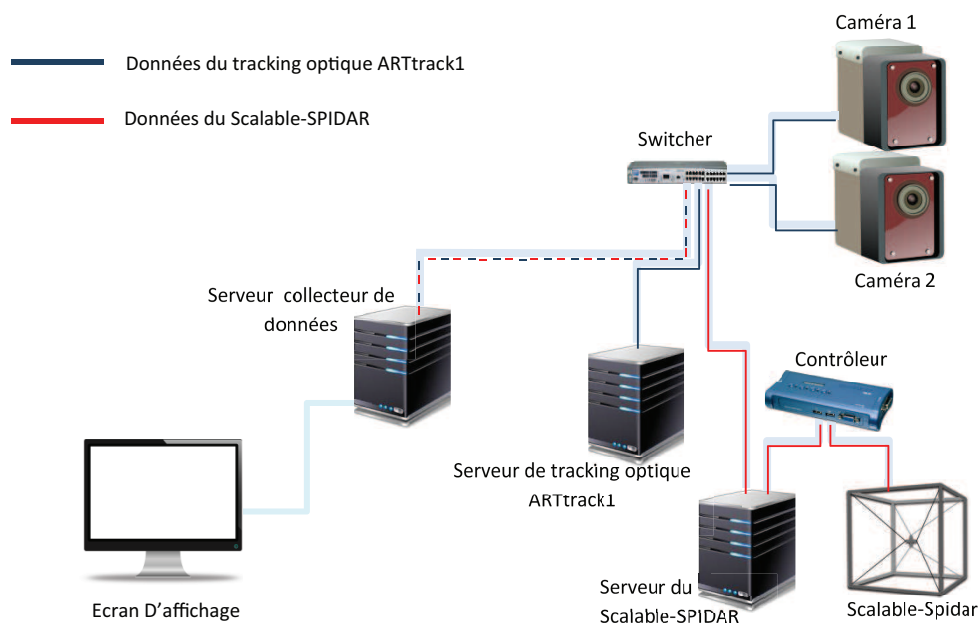


FIGURE 4.10 – Architecture du protocole proposé

Comme le montre la figure 4.10, l'architecture employée s'articule autour de cinq éléments :

- Un logiciel DTrack[®] qui permet de calculer et communiquer les données capturées par les deux caméras infrarouges
- Un serveur de tracking optique qui, une fois sollicité, envoie continuellement des trames contenant des informations sur les identifiants et les positions des cibles traquées.

- Un serveur du Scalable-SPIDAR qui à son tour, une fois sollicité, envoie continuellement des trames contenant des informations sur la position et l'orientation de l'effecteur ainsi que les valeurs provenant des différents encodeurs.
- Un serveur Collecteur de Données qui, une fois lancé, associe les données provenant du Scalable-SPIDAR et du serveur de tracking optique afin de les exploiter ultérieurement pour la création des périphériques virtuels.
- Un logiciel Virtools qui permet une mise en œuvre rapide des applications de réalité virtuelle (Voir annexe A).

L'originalité de notre protocole est que celui-ci utilise la réalité virtuelle pour caractériser le Scalable-SPIDAR. Ceci implique qu'il est important de créer une application de réalité virtuelle ou encore une composition qui soit simple à utiliser, à mettre en place et à administrer.

Dans la suite, nous décrivons brièvement la structure de la composition créée avec Virtools. Cette composition s'articule autour de trois scripts de comportement principaux.

Le premier script est dédié à la génération d'une scène virtuelle constituée d'un ensemble de petits cubes où chaque cube représente un sous espace de l'espace de travail du Scalable-SPIDAR. L'ensemble de ces petits cubes forme un grand cube qui couvre une zone bien déterminée de l'espace de travail. Le grand avantage de ce script est qu'il permet de couvrir un espace de travail fixé et contrôlé par l'utilisateur. Celui-ci a donc la possibilité de modifier de manière directe les dimensions de l'espace de travail ainsi que la granularité des nuages de points.

Le deuxième script gère les interactions entre l'utilisateur et la scène virtuelle déjà générée par le premier script et assure l'enregistrement des informations venant des périphériques de tracking en cas de collision détectée.

Ce script implémente un guide visuel pour faciliter la tâche de sélection. Celui-ci permet de créer un guidage visuel vers les objets d'intérêt dans la scène virtuelle soient les petits cubes. En effet, le guide affiche la mesure à saisir en colorant le cube correspondant en rose. La désignation du cube coloré s'effectue de façon incrémentale grâce à un module mis en place **Calculate IC** qui retourne l'indice du cube à saisir (figure 4.11).

Une fois le cube désigné est sélectionné, nous ferons varier la couleur du cube qui devient verte. Ce guide d'aide à la sélection contribue à la mise en œuvre d'une méthode de collecte de données simple et maîtrisée en accord avec le caractère quasi-statique de notre protocole. De fait, même un déplacement brusque de l'effecteur n'a aucune influence sur les autres cubes vu que l'utilisateur ne peut valider que le cube affiché. Ceci permet entre autre de garantir l'unicité de données.

Afin de contraindre les mesures à saisir à des positions plus précises et reproductibles, nous nous sommes servis d'un module de détection de collision prédéfini (figure 4.12). Ce module a pour rôle de détecter toute collision entre l'effecteur virtuel et le cube affiché.

Or, comme nous l'avons expliqué précédemment, une détection de collision simple ne résout pas le problème. Pour cela, nous avons opté pour la détection d'une double collision. Une solution astucieuse consiste à inscrire à l'intérieur de chaque cube autour de son centre une petite sphère de faibles dimensions. Ceci nous permettra de nous approcher le plus possible des centres des cubes.

Ainsi, la première collision détectée indique que l'effecteur virtuel modélisé par une sphère vient de pénétrer à l'intérieur du cube sélectionné. Au contraire de l'ancien protocole, l'enregistrement de données renvoyées par le Scalable-SPIPAR et le système ART-track1 n'est lancée que suite à la détection d'une deuxième collision entre la sphère ins-

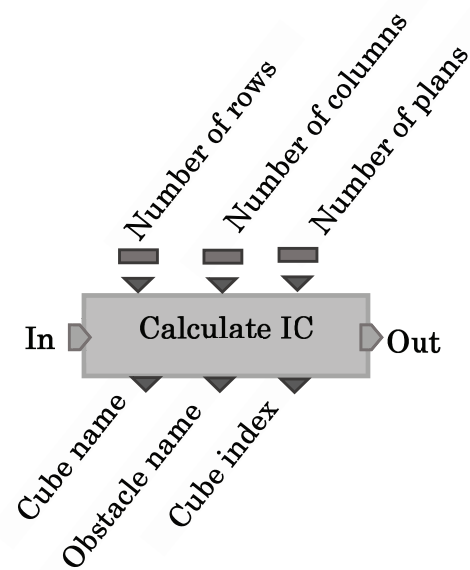


FIGURE 4.11 – Le module Calculate IC

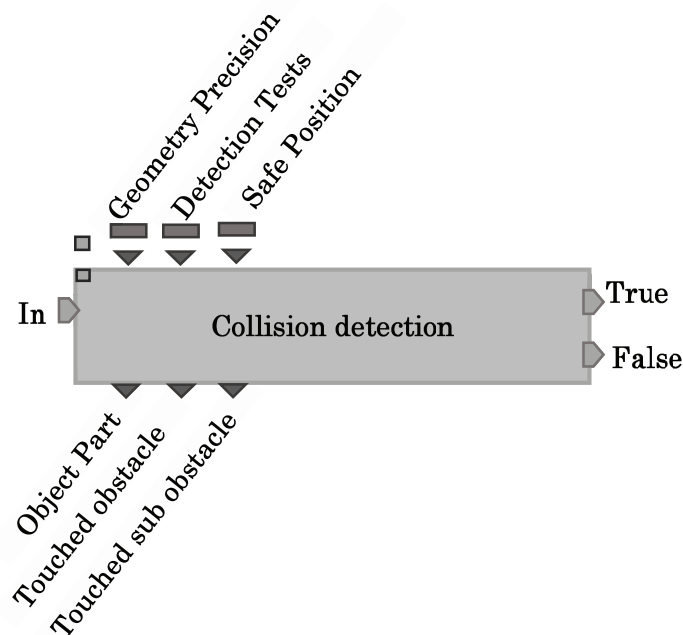


FIGURE 4.12 – Le module collision detection

crité au centre du cube est permet déclencher l'enregistrement des informations. Ces informations sont stockées au fur et mesure dans un tableau et exploitées ensuite pour la caractérisation des erreurs du Scalable-SPIDAR.

Enfin, le troisième script s'occupe de la récupération des informations qui proviennent du le Scalable-SPIADR et du Système ARTtrack1. Ce script repose sur une bibliothèque de modules personnalisés ou building blocks qui a été mise en place. Le premier mo-

Le module **DCCU Tracker** permet de recueillir les informations venant d'un système de tracking telles que la position et l'orientation (figure 4.13).

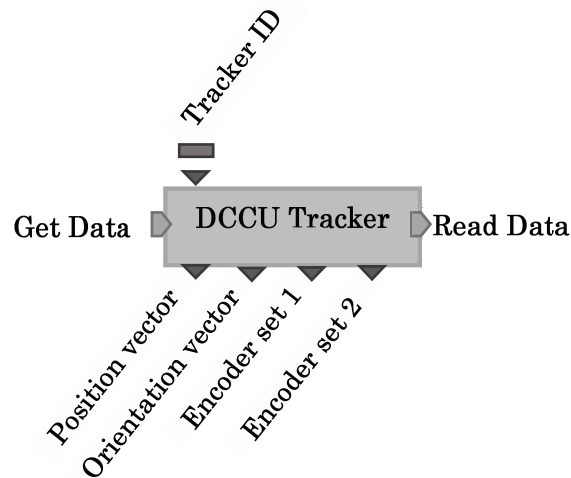


FIGURE 4.13 – Le module DCCU Tracker

Dans notre cas, ces informations peuvent provenir du Scalable-SPIDAR ou du système de tracking optique ARTtrack1 via DTRack[®]. Nous choisissons les périphériques à écouter en passant en paramètres leurs identifiants (Tracker ID). Le deuxième module **DCCU Init** permet d'initialiser la communication avec les différents périphériques à écouter (figure 4.14).

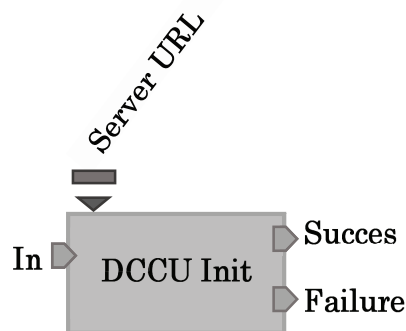


FIGURE 4.14 – Le module DCCU Init

Dans la présente section, nous avons brossé les grandes lignes de la composition. Pour plus de détails sur ce sujet, veuillez consulter l'annexe B.

4.3.2.3 Déroulement du nouveau protocole

Dans cette section, nous présentons brièvement le déroulement de notre protocole.

1. Un guide visuel indique à l'utilisateur la mesure à saisir en colorant le petit cube correspondant en rose. L'utilisateur déplace l'effecteur dans l'espace afin de placer un effecteur virtuel dans le cube désigné par le système (figure 4.15).

2. A chaque fois que l'effecteur virtuel entre en collision avec le cube désigné, le cube change de couleur et devient vert pour confirmer la sélection. Une deuxième détection de collision est indispensable pour pouvoir enregistrer la position renvoyée par le Scalable-SPIDAR, la position renvoyée par le système de tracking optique ARTtrack1, le numéro du cube ainsi que les valeurs venant des différents encodeurs.
3. Une fois ces informations sont récupérées, le cube sélectionné disparaît et le guide visuel indique de nouveau le prochain cube à saisir garantissant ainsi l'unicité des données. Ce processus est répété jusqu'à ce que l'utilisateur termine la saisie de tous les cubes constituant la représentation virtuelle de l'espace de travail étudié.

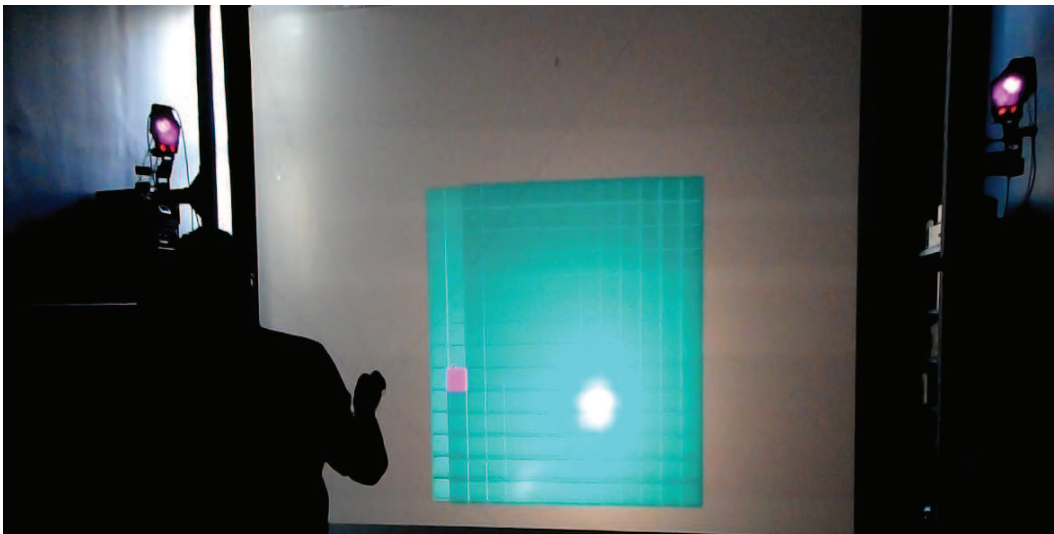


FIGURE 4.15 – Cadre expérimental du nouveau protocole. Un guide visuel d'aide à la sélection indique à l'utilisateur le cube à saisir.

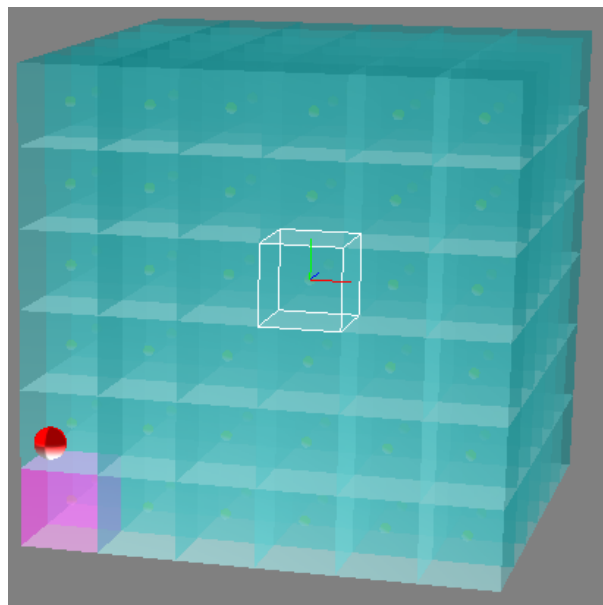


FIGURE 4.16 – Représentation virtuelle de l'espace de travail : Chaque cube est muni d'une sphère autour de son centre.

4.3.2.4 *Bilan de notre protocole de collecte de données*

Le développement d'un protocole expérimental permettant la caractérisation du Scalable-SPIDAR a été difficile, dans la mesure où il est très difficile de caractériser un système de tracking. Notons que cette volonté de proposer un protocole précis et fiable se traduit dans le fait qu'il est le fruit d'échanges collaboratifs entre les membres de notre équipe de recherche. Nous pensons utile de préciser les avantages et les inconvénients du protocole conçu.

Premièrement, ce protocole est innovant dans la mesure où, nous avons utilisé les techniques de réalité virtuelle pour récupérer les positions renvoyées par le Scalable-SPIDAR et le système ARTtrack1 dans un espace de travail défini au préalable de manière homogène et unique. L'idée originale est de se servir des guides virtuels d'aide à la sélection pour assister l'utilisateur à placer son effecteur dans l'espace de façon à récupérer un ensemble de positions qui seront réparties dans l'espace de travail. Il en résulte non seulement une interaction facilitée mais aussi une plus grande efficacité pour effectuer la tâche de collecte de données. En effet, grâce à ces guides nous arrivons à interdire les déplacements brusques et rudes de l'effecteur et par conséquent aboutir à une interaction fluide et continue. La collecte de données est alors effectuée de façon guidée et sans subir de discontinuité. Ceci a pour conséquence d'augmenter la qualité de données récoltées dans la mesure où il nous permet de s'affranchir des situations menant à des sauts ou redondance de données. En outre, l'utilisation de ces guides offre un gain de facilité et permet ainsi la mise en œuvre d'un protocole suffisamment léger et simple qui ne requière aucune période d'apprentissage.

Deuxièmement, ce protocole est facile à administrer, dans la mesure où, l'utilisateur est capable de contrôler facilement les propriétés de la représentation virtuelle de l'espace de travail. Ainsi, pour agrandir ou réduire la taille de l'espace de travail, il suffit juste d'agir sur les paramètres de dimensions. Il est possible également de faire varier la densité des mesures à l'intérieur de l'espace de travail en modifiant la taille du cube élémentaire. Ceci permet à l'utilisateur d'obtenir une représentation riche et fidèle de l'espace de travail qu'il envisage calibrer.

Troisièmement, ce protocole a le grand avantage d'être robuste. Il peut être facilement transposé à d'autres systèmes de tracking en particulier ceux possédant un grand espace de travail. En effet, suivant les dimensions du système de tracking à caractériser, l'utilisateur ajoute les propriétés de la représentation virtuelle afin de faire coïncider celle-ci sur l'espace de travail voulu.

Quatrièmement, ce protocole se distingue par sa capacité à contraindre les mesures à saisir à des positions précises. L'ajout d'une petite sphère à l'intérieur de chaque petit cube (figure 4.16) a permis de supprimer les éventuels décalages entre les mesures récoltées et les centres de cubes et par conséquent garantir la reproductibilité de l'expérimentation.

Pour plus de clarté, nous présentons deux séquences de collecte de données effectuées respectivement à l'aide de l'ancien et du nouveau protocole. La séquence est composée de 4096 points.

La figure 4.17 souligne le caractère aléatoire et non régulier de la première méthode de collecte de données adoptée par Pierre Boudoin [BOUDOIN \[2010\]](#). Notons bien, les sauts de données dues à des déplacements brusques de l'effecteur ainsi que la redondance de quelques mesures.

La figure 4.18 illustre la séquence de collecte de données effectuée par le nouveau protocole.

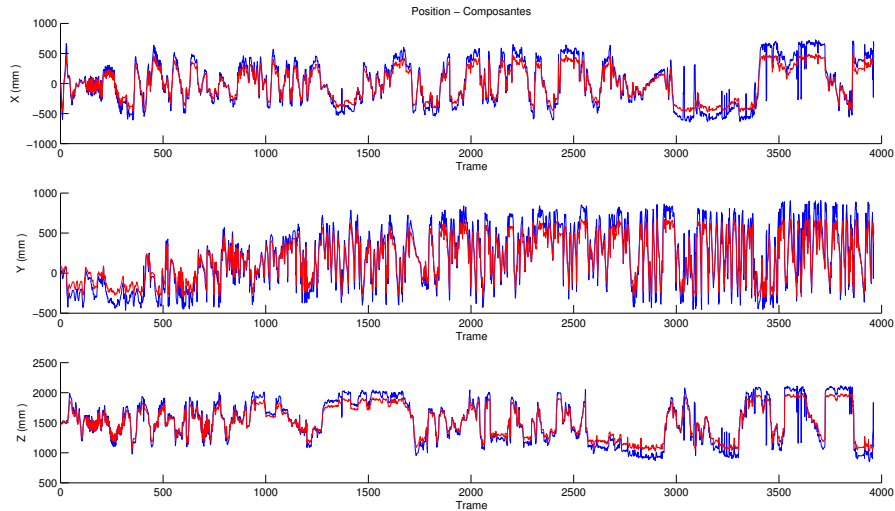


FIGURE 4.17 – Ancien protocole de collecte de données

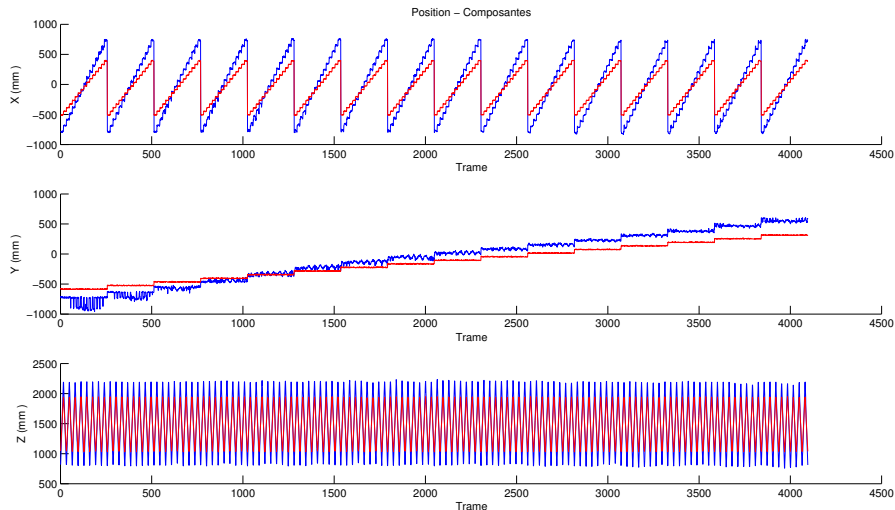


FIGURE 4.18 – Nouveau protocole de collecte de données

Les mesures sont prises de manière incrémentale dans le même plan xz . Une fois l'utilisateur termine la saisie de tous les cubes constituant ce plan, il avance dans la direction $+Y$ pour effectuer la saisie du plan suivant. Cette nouvelle stratégie permet une collecte de données incrémentale et régulière dans l'espace de travail. Elle est en accord avec le caractère quasi-statique du protocole et interdit implicitement les déplacements brusques de l'effecteur et par conséquent les sauts de données. Notons enfin qu'avec ce protocole, nous sommes arrivées à saisir les 4096 mesures et donc garantir l'unicité des mesures relevées.

Les figures 4.19 et 4.20 démontrent la capacité de notre protocole à contraindre les mesures à saisir à des positions précises et par conséquent garantir la reproductibilité de l'expérimentation. La figure 4.19 représente notre espace de travail idéal qui peut être vue comme un grand cube situé à l'origine du Scalable-SPIDAR tandis que la figure 4.20 représente notre espace de travail reconstruit à partir des mesures relevées lors d'une campagne de mesures. L'espace décrit par les mesures est identique à l'idéal et présente

quelques légères différences.

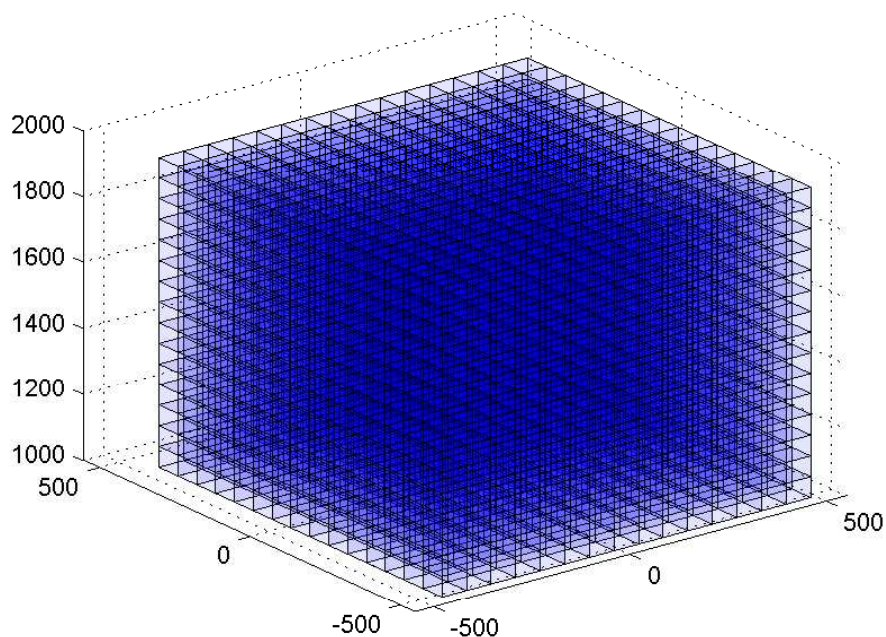


FIGURE 4.19 – Représentation virtuelle idéale de notre espace de travail

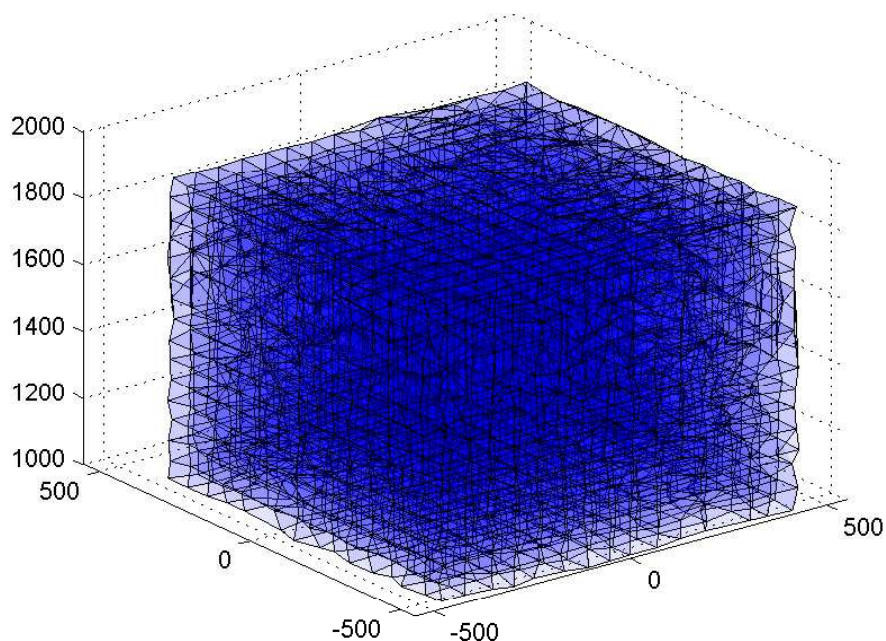


FIGURE 4.20 – Reconstruction de notre espace de travail à partir des mesures relevées lors d'une campagne de mesure

Malheureusement, malgré tous les avantages annoncés ci-dessus, ce protocole demeure long et fastidieux surtout si le nombre de mesures à récupérer est très élevé. Après un certain temps d'utilisation, celui-ci peut être aussi une source d'inconfort et de fatigue.

CONCLUSION

Dans ce chapitre, nous avons identifié les différents problèmes rencontrés dans le Scalable-SPIDAR. Pour pallier ceux-ci nous avons proposé une initialisation multimodale permettant d'assister l'utilisateur durant la phase d'initialisation avec une très grande précision. Ensuite, nous avons mis en place un protocole de collecte de données rigoureux et fiable afin de caractériser et souligner l'influence des problèmes de conception sur la précision des données de position du Scalable-SPIDAR. Ce protocole original s'appuie sur les techniques de la réalité virtuelle pour effectuer la collecte des données de calibration. Les critères de fiabilité, validité, robustesse ainsi de commodité sont les points forts du protocole mis en place. Dans le prochain chapitre, nous exploitons les données collectées pour développer des méthodes de calibration adéquates.

Chapitre 5

Caractérisation et calibration du Scalable-SPIDAR

Résumé

Nous présentons dans ce chapitre la deuxième phase de la procédure de calibration du Scalable-SPIDAR. Cette phase propose l'utilisation de deux techniques de régression non-linéaires : les réseaux de neurones et la régression à vecteurs de support pour la calibration. Ces deux techniques se distinguent par leurs qualités d'approximateurs universels. Elles visent à améliorer la précision des données de position du Scalable-SPIDAR par rapport à une référence fournie par un système de tracking optique ARTtrack1 plus précis. Dans une deuxième section, nous développons deux prototypes d'applications de réalité virtuelle afin de mettre en évidence l'apport des méthodes de traitement de données.

Sommaire

Introduction	117
5.1 Hypothèses de travail (Volume utile)	117
5.2 Interprétation des résultats de caractérisation	118
5.2.1 Introduction	118
5.2.2 Incertitudes de position du Scalable-SPIDAR	118
5.2.3 Etude de la répétabilité	120
5.2.4 Comparaison avec les travaux de Pierre Boudoin	122
5.3 Calibration du Scalable-SPIDAR à l'aide des techniques de régression non linéaires	123
5.3.1 Calibration du Scalable-SPIDAR à l'aide des réseaux de neurones	124
5.3.1.1 Prétraitement des données	124
5.3.1.2 L'approche expérimentale	126
5.3.1.2.1 Configuration et optimisation des caractéristiques du réseau de neurones	126
5.3.1.2.2 Performances du réseau de neurones en apprentissage	128
5.3.1.2.3 Performances du réseau de neurones en généralisation	131

5.3.2	Calibration du Scalable-SPIDAR à l'aide des SVR	133
5.3.2.1	Performances des SVR en apprentissage	134
5.3.2.2	Performances des SVR en généralisation	135
5.3.3	Mise en balance des méthodes de calibration	136
5.4	Mise en pratique de la procédure de calibration du Scalable-SPIDAR . . .	137
5.5	Applications pratiques	140
5.5.1	Modélisation 3D	140
5.5.1.1	Contraintes architecturales	140
5.5.1.2	Techniques de modélisation	140
5.5.1.3	Organisation des modèles 3D	142
5.5.2	L'Animation 3D	144
5.5.2.1	Les contraintes fonctionnelles	144
5.5.2.2	Techniques d'animation 3D	145
5.5.2.3	Squelettes des modèles 3D	146
5.5.3	Prototypes d'applications de réalité virtuelle	147
5.6	Les pistes de notre action	147
Conclusion	149

INTRODUCTION

Ce chapitre traite la deuxième phase de la conception d'une procédure de calibration d'une interface à retour d'effort à échelle humaine : le Scalable-SPIDAR. La première phase a consisté à l'élaboration d'un cadre expérimental puis à la mise en place d'un protocole permettant la caractérisation du périphérique. Typiquement, celui-ci sert à mettre l'accent sur l'influence des problèmes inhérents au périphérique sur la précision des données de position. Pour cette deuxième phase nos contributions sont premièrement la caractérisation des erreurs statiques de position du Scalable-SPIDAR. Celle-ci est une étape clé pour pouvoir choisir la meilleure méthode pour calibrer le périphérique en question. La deuxième contribution est le développement d'une méthode de calibration appropriée permettant de corriger la position renvoyée par le Scalable-SPIDAR sans avoir à priori une connaissance de son modèle mathématique. Une approche de calibration boîte noire est proposée sous la forme d'une étude comparative entre deux techniques de régression non linéaires : les réseaux de neurones et la régression à vecteurs de support.

5.1 HYPOTHÈSES DE TRAVAIL (VOLUME UTILE)

Afin d'obtenir une bonne caractérisation du Scalable-SPIDAR, nous préconisons l'utilisation d'un volume utile plus petit que celui offert en théorie. Ainsi, d'un volume théorique de $27m^3$ nous passons à un volume de $1m^3$.

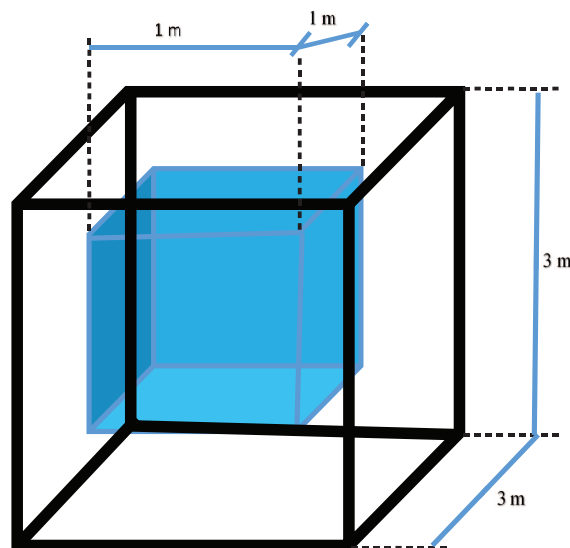


FIGURE 5.1 – Représentation du volume utile en bleu du Scalable-SPIDAR à l'intérieur de sa structure (en noir).

Cette restriction nous permet d'une part, de concentrer nos efforts sur le volume le plus souvent exploité et d'autre part, d'éviter les situations dans lesquelles le Scalable-SPIDAR opère au bout de ses limites. En effet, hors les limites de ce volume, la précision diminue drastiquement vu la quantité importante de câble qu'il faut enrouler et les mesures retournées sont donc susceptibles d'être faussées.

5.2 INTERPRÉTATION DES RÉSULTATS DE CARACTÉRISATION

5.2.1 Introduction

Avant de commencer l'interprétation des résultats de caractérisation, nous présentons quelques notions mathématiques pour le calcul des composantes des erreurs.

La **position moyenne** p_{moy} de N positions enregistrées p_i $i = 1, \dots, N$ peut être calculée en faisant la moyenne de N vecteurs de positions : $p_{moy} = \frac{1}{N} \sum_{i=1}^N p_i$

La **distance euclidienne** d entre deux points $x = (x_1, x_2, x_3)^T$ et $y = (y_1, y_2, y_3)^T$ s'écrit :

$$d(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^3 (x_i - y_i)^2} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2} \quad (5.1)$$

Le vecteur d'erreur est défini par :

$$\vec{err}_i = \vec{r}_{r_i} - \vec{r}_{m_i} = \begin{pmatrix} x_{référence} \\ y_{référence} \\ z_{référence} \end{pmatrix} - \begin{pmatrix} x_{mesuré} \\ y_{mesuré} \\ z_{mesuré} \end{pmatrix} \quad (5.2)$$

Avec r_{r_i} est la position de référence à laquelle l'entité traquée est censée être, r_{m_i} est la position mesurée par le système de tracking.

L'erreur absolue de position (err_{pos}) est définie comme la distance entre la position réelle p_{cou} dans l'espace et la position mesurée par le système de tracking. Pour la déterminer, nous avons besoin d'une valeur sûre $p_{réf}$ issue d'un deuxième système de tracking plus précis. L'erreur absolue de position s'écrit alors : $err_{pos}(x) = d(p_{réf}, p_{cou}) = \|p_{réf} - p_{cou}\|_2$

5.2.2 Incertitudes de position du Scalable-SPIDAR

Afin de mettre en évidence les erreurs statiques de position, nous avons choisi de représenter la distribution d'erreur absolue sous trois angles supplémentaires, c'est-à-dire :

- Un graphique en 3D représentant la distribution spatiale de l'erreur absolue à chaque point de référence (x_r, y_r, z_r) . Notons que la couleur de l'erreur est proportionnelle à son amplitude.
- Un graphique en 1D représentant les erreurs absolues dans l'ordre dans lequel ont été collectées.
- Un diagramme en boîte représentant les erreurs signées relatives à chacun des axes.
- L'histogramme des erreurs absolues de position.
- La distribution de l'erreur absolue en fonction de la distance à laquelle la mesure a été prise du centre du périphérique.

Ces différents graphiques constituent une base solide permettant une bonne caractérisation de notre périphérique : le Scalable-SPIDAR. Comme le montre la figure 5.2, la distribution de l'erreur absolue dans l'espace de travail du Scalable-SPIDAR est de type "pelure d'oignon" c'est-à-dire plusieurs couches, plutôt sphériques, l'erreur absolue s'accroissant au gré des couches.

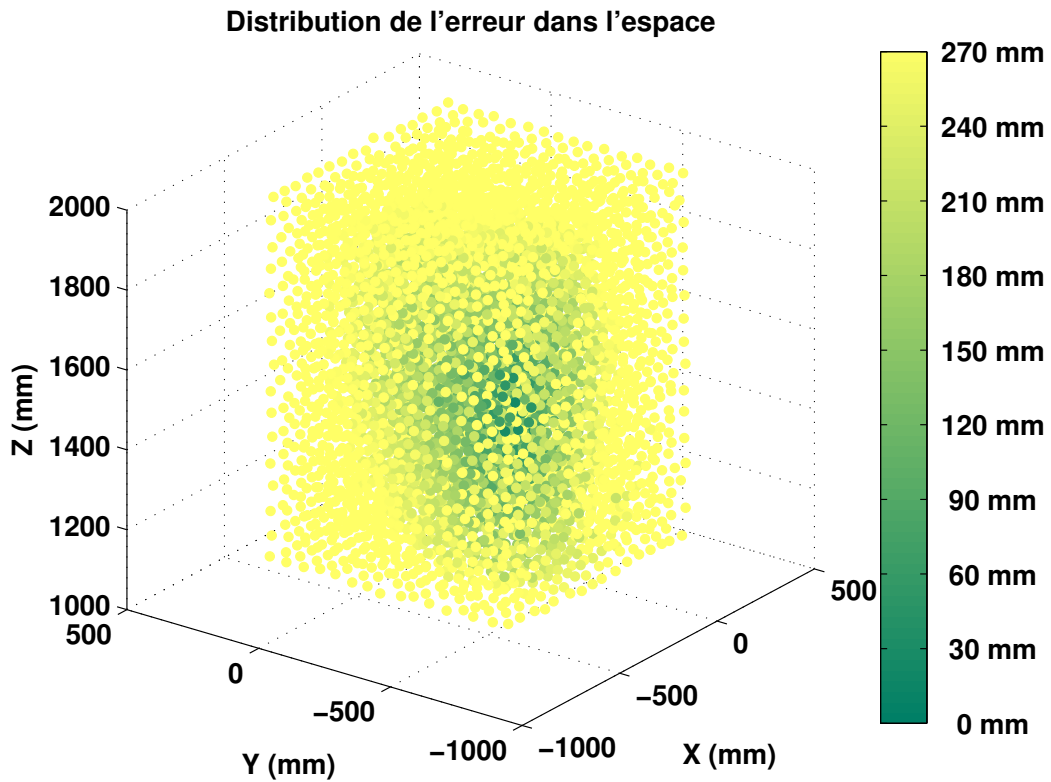


FIGURE 5.2 – Répartition de l'erreur absolue de position actuelle dans l'espace de travail

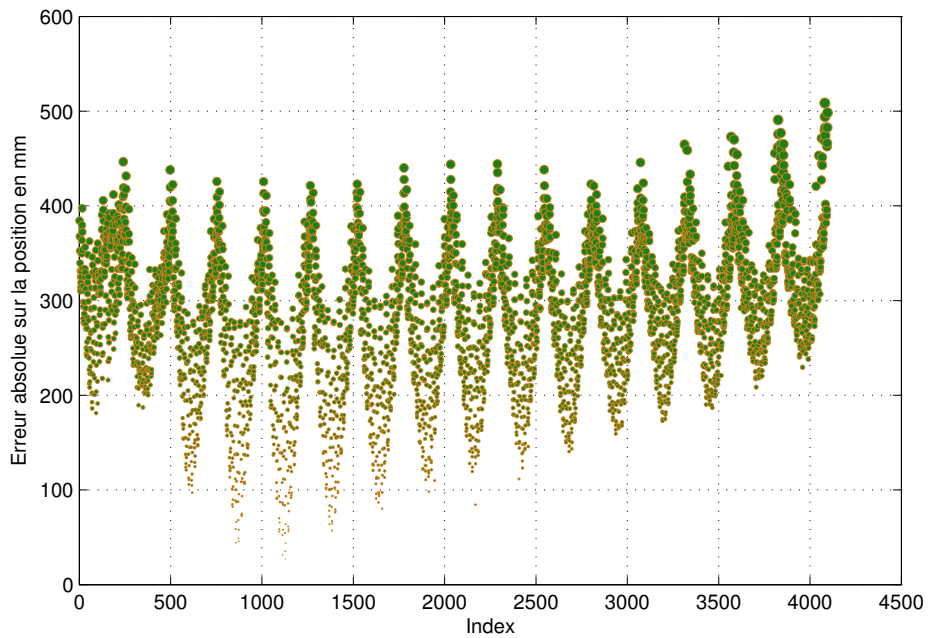


FIGURE 5.3 – Erreur absolue représentée en séquence

Les régions où l'erreur de position est la plus conséquente, sont situées sur les différents coins de l'espace de travail.

La figure 5.3 représente les erreurs absolues dans l'ordre dans lequel ont été collectées. La périodicité de la figure nous permet de conclure que les erreurs de position les plus im-

portantes sont situées sur le dessus et sur les coins de l'espace de travail. Le diagramme en boîte (figure 5.4) présente l'erreur pour les trois coordonnées spatiales ainsi que la résultante correspondante à la position absolue de l'effecteur. Les calculs ont été menés à partir des 4096 points de calibration. Les trois colonnes à gauche représentent la distribution de l'erreur selon chaque composante. La représentation est simple dans la mesure où l'erreur est décomposée en trois composantes : x , y et z . Notons que les composantes sont des erreurs signées et que l'erreur absolue est non signée.

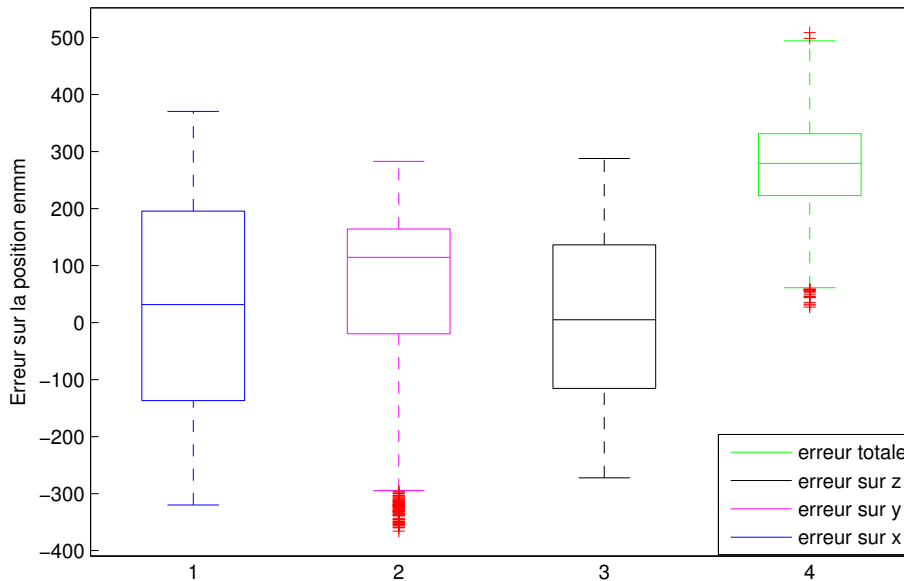


FIGURE 5.4 – Diagramme en boîte de l'erreur de position

Nous constatons une certaine hétérogénéité de l'erreur suivant les trois dimensions. Cette remarque est surtout vraie pour la composante Y. Afin de bénéficier d'informations complémentaires sur la distribution de l'erreur absolue, celle-ci est représentée par un histogramme.

Il est clair que la distribution de l'erreur absolue est non normale (figure 5.5). Ce type de distribution n'est pas surprenant de fait que les données analysées sont des distances euclidiennes qui sont par définition positives. En effet, nous sommes face à un histogramme étalé, centré autour d'une moyenne de 274,82 mm et d'écart type de 78,12 mm qui témoigne d'une mauvaise précision et d'une grande dispersion statistique.

5.2.3 Etude de la répétabilité

Cette étude consiste à observer les différentes valeurs de position renvoyées par le Scalable-SPIDAR pour une même position de la représentation virtuelle de l'espace. **L'erreur de répétabilité** ($err_{répétabilité}$) est définie comme la racine carrée de la distance moyenne quadratique à la position moyenne :

$$err_{répétabilité} = \sqrt{\frac{1}{N} \sum_{i=1}^N d(p_{moy}, p_i)^2} \quad (5.3)$$

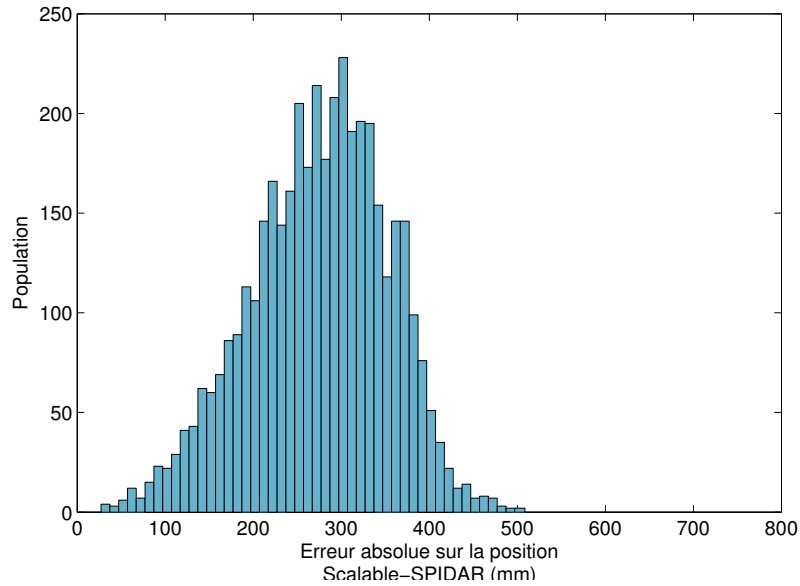


FIGURE 5.5 – Histogramme de la distribution d'erreur absolue de position

Pour cela, quatre campagnes de mesures ont été effectuées à la même position en se basant sur le protocole déjà conçu. Nous avons relevé des mesures en respectant le protocole mis en place avec un volume de travail utile de 1 m^3 décomposé en 2197 cubes. Les données couvrent bien la globalité du volume de travail en trois dimensions.

Il est clair que l'erreur de répétabilité est sensible à la distance par rapport à laquelle la mesure a été prise du centre du Scalable-SPIDAR. En effet, si l'effecteur est placé à la limite du volume utile l'erreur augmente progressivement. Les mesures sont plus répétables si nous restons proche du centre du Scalable-SPIDAR.

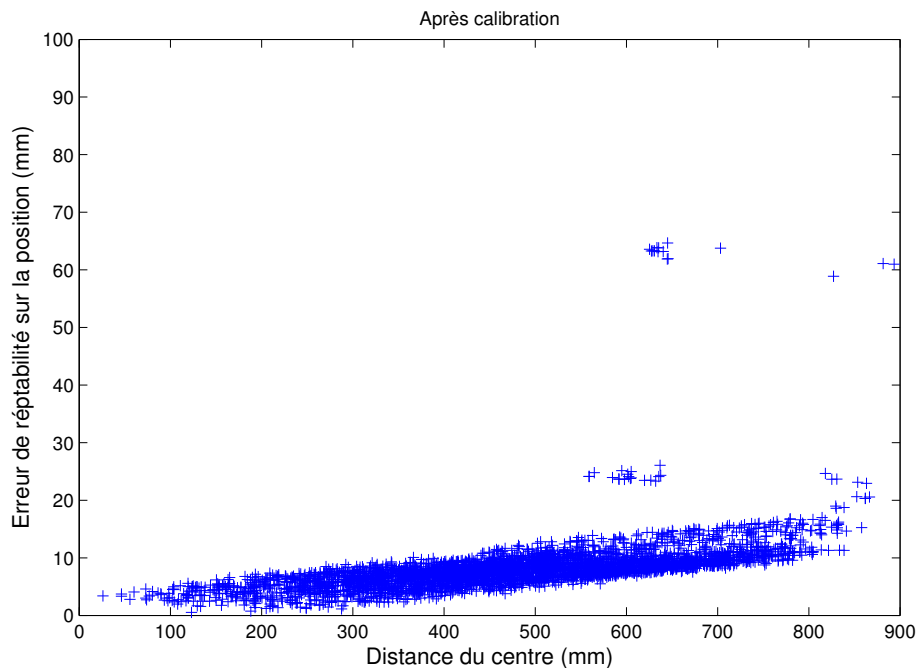


FIGURE 5.6 – Étude de l'erreur de répétabilité

5.2.4 Comparaison avec les travaux de Pierre Boudoin

Comme nous l'avons d'ores et déjà mentionné, notre travail de thèse s'inscrit dans le cadre d'une continuité d'un travail antérieur mené par Pierre Boudoin [BOUDOIN \[2010\]](#) et portant aussi sur l'étude et la calibration du Scalable-SPIDAR. Une comparaison entre la distribution d'erreur relevée et celle décrite dans les travaux de Pierre Boudoin nous semble indispensable pour aboutir à des solutions pertinentes et cohérentes avec la situation actuelle de notre périphérique : le Scalable-SPIDAR.

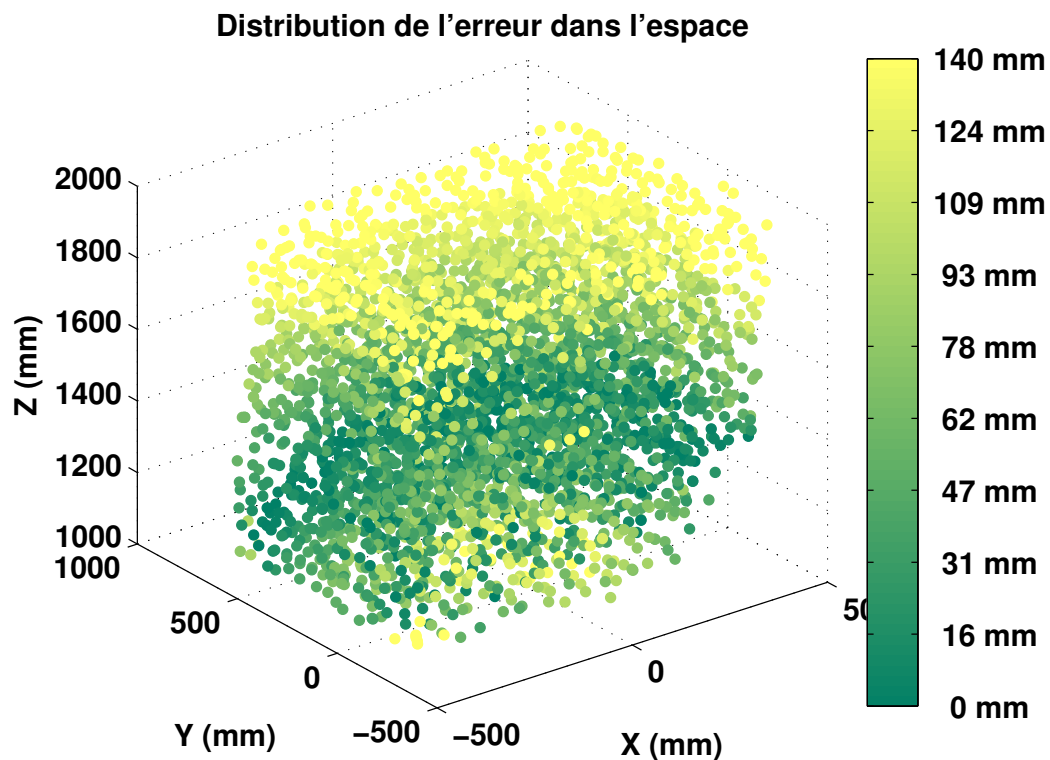


FIGURE 5.7 – Répartition de l'erreur absolue de position dans l'espace de travail relevée par Pierre Boudoin

Comme nous pouvons le remarquer dans la figure 5.7, la distribution de l'erreur absolue relevée par Pierre Boudoin est aussi de type "pelure d'oignon". L'erreur augmente au fur et à mesure que nous nous éloignons l'effecteur de l'origine du Scalable-SPIDAR. Cependant, nous constatons que l'ordre de grandeur de l'erreur absolue n'est pas le même que celui décrit ci-dessus. En effet, nous remarquons que l'erreur a augmenté drastiquement dans notre situation en particulier sur les coins de l'espace de travail. Pour une meilleure comparaison, les principales valeurs caractéristiques de la distribution d'erreur absolue sur l'espace de travail du Scalable-SPIDAR obtenue et celle décrite dans les travaux de Pierre Boudoin sont présentées dans le tableau 5.1.

TABEAU 5.1 – Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR

Erreur absolue	Décrite dans les travaux de Pierre Boudoin	Actuelle
Moyenne (mm)	72.86	274.82
Ecart type (mm)	47.06	78.12
Erreur maximale (mm)	211.29	508.56

Le tableau 5.1 représente la valeur moyenne, l'écart type et la valeur maximale de l'erreur absolue. Les calculs statistiques ont été menés à partir de 4096 points. Nous constatons à la lecture du tableau, que l'ordre de grandeur des erreurs absolues dans notre situation est bien supérieur à celui décrit dans les travaux de Pierre Boudoin. L'erreur a augmenté de façon drastique pour atteindre $274.82mm$.

Nous pouvons nous interroger sur les raisons de ces résultats notamment très différents de ceux avancés par Pierre Boudoin. Plusieurs causes peuvent être avancées. Premièrement, il apparaît clairement que le Scalable-SPIDAR a subi une dégradation considérable. Le firmware du contrôleur HDHC (High Definition Haptic Controller) souffre d'un bug qui dégrade la qualité de contrôle. D'autre part, la structure mécanique du Scalable-SPIDAR a été remise en cause et des nouveaux blocs moteur-encodeurs ont été conçus. Bien que le problème de frottement soit résolu grâce aux nouveaux blocs, les erreurs de position demeurent. L'interface est toujours incapable de mesurer la longueur exacte des câbles. La méthode pour mesurer la longueur des câbles à travers le nombre de tours des moteurs est inefficace vu que le rayon du câble sur le moteur continue à changer.

En effet, cette situation semble être une bonne opportunité pour démontrer la pertinence et le bien fondé de notre approche pour la calibration du Scalable-SPIDAR. Les études précédentes ont montré que les données de position sont entachées d'erreur statique sur tout le volume de travail et qui restent trop élevées en particulier sur les différents coins. Nous savons dès maintenant que nous ne pouvons pas échapper à une correction des erreurs et qu'en aucun cas ce type de périphérique ne peut être utilisé sans phase de calibration. Il s'agit, dès lors, de corriger ces erreurs à l'aide des méthodes de traitement de données.

5.3 CALIBRATION DU SCALABLE-SPIDAR À L'AIDE DES TECHNIQUES DE RÉGRESSION NON LINÉAIRES

Comme nous l'avons expliqué dans le chapitre précédent, l'interface du Scalable-SPIDAR est une boîte noire. De fait, nous n'avons aucune connaissance a priori du modèle permettant de déterminer la position de l'effecteur du Scalable-SPIDAR et donc nous ne pouvons en aucun cas agir là dessus. Deux solutions semblent se profiler. La première solution requiert un modèle analytique décrivant la correspondance entre les coordonnées articulaires et les coordonnées de l'effecteur. Ce modèle fait intervenir un certain nombre de paramètres liés à la géométrie du Scalable-SPIDAR (modèle à représentation physique). La deuxième solution consiste à utiliser une approche de calibration boîte noire. En effet, il est possible d'obtenir un modèle pouvant établir la correspondance directe entre les coordonnées entachées d'erreurs et de références avec la précision escomptée sans l'aide du modèle géométrique du périphérique, c'est-à-dire même en ne connaissant pas a priori la géométrie du périphérique à calibrer. La deuxième approche nous semble être en accord avec l'investigation que nous avons menée précédemment. En effet, le Scalable-SPIDAR souffre des problèmes dont nous ne faisons que soupçonner la cause et pour lesquels nous avons du mal à évaluer l'influence sur la précision des données de position. Il s'agit dès lors de modéliser la fonction de régression qui relie les positions données par le Scalable-SPIDAR considérées comme entachées d'erreur aux positions données par le système de tracking optique ARTtrack1 considérée comme références.

A la lumière des remarques précédentes, nous estimons louable l'utilisation des tech-

niques de régression non linéaires et plus précisément les réseaux de neurones et les machines à vecteurs de support pour obtenir une bonne approximation de la fonction de régression. En effet, ces deux techniques possèdent une propriété très intéressante qui est à l'origine de leur intérêt : ce sont des approximateurs universels parcimonieux.

5.3.1 Calibration du Scalable-SPIDAR à l'aide des réseaux de neurones

Pour être en mesure d'utiliser les réseaux de neurones, nous devons dans un premier temps préparer les données qui se présenteront en entrée et en sortie. Les étapes qu'il ne faut pas les prendre à la légère sont les étapes de prétraitement. Les étapes de prétraitement améliorent d'une façon considérable la précision de la régression.

5.3.1.1 Prétraitement des données

Le prétraitement de données est une étape cruciale car elle nous permet de sélectionner, dans l'espace de représentation, les informations nécessaires à l'application. Cette sélection passe par la normalisation des données ainsi que la suppression des redondances. Les données subissant un prétraitement dans ce travail de recherche sont les positions renvoyées par le Scalable-SPIDAR et le système de tracking optique ARTtrack1.

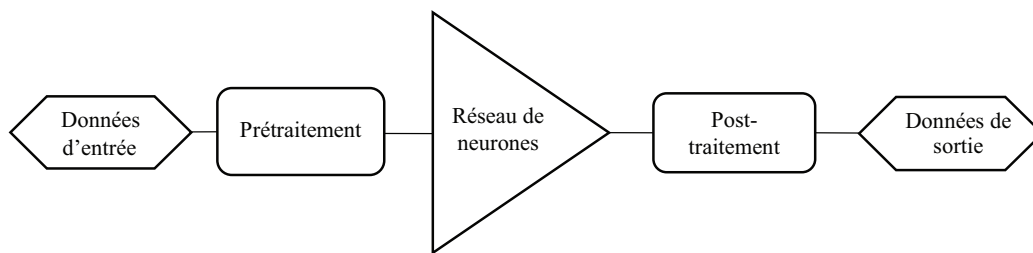


FIGURE 5.8 – Chaîne de prétraitement de données

La figure 5.8 décrit un système simplifié utilisé dans les applications de régression à base de réseau de neurones. La première de la chaîne de prétraitement est d'effectuer une normalisation des données suivie d'un centrage. Grâce à ce type de traitement, l'apprentissage du réseau de neurones est facilité puisque les données d'entrée ainsi que celles de sortie possèdent toutes le même ordre de grandeur. Notons les symboles suivants ainsi que leurs définitions :

N est le nombre de points de mesure récoltés

q désigne les trois de coordonnées spatiales

x^k est le vecteur de position pour le $k^{i\grave{e}me}$ point

x_i^k est la valeur de l' $i^{i\grave{e}me}$ coordonnée pour le $k^{i\grave{e}me}$ point

Les données normalisées sont obtenues à partir des données brutes selon les équations suivantes :

$$x_i^{k'} = (\Omega_i x_i^k + \Sigma_i) \quad (5.4)$$

Où

Ω_i : facteur multiplicatif de la coordonnée i

Σ_i : facteur additif de la coordonnée i

Pour calculer les valeurs normalisées à partir de l'équation ci-dessous, nous devons d'abord déterminer les valeurs moyennes et les écarts types pour les entrées et sorties.

Les moyennes sont fournies par :

$$M_i = \frac{\sum_k x_i^k}{N} \quad (5.5)$$

Et les écarts types sont fournis par les relations suivantes :

$$\sigma_i = \sqrt{\frac{\sum_k (x_i^k)^2}{N} - \frac{\left(\sum_k x_i^k\right)^2}{N^2}} \quad (5.6)$$

Les facteurs multiplicatifs et additifs sont définis comme suit :

$$\Omega_i = \frac{1}{\sigma_i} \text{ et } \Sigma_i = -\frac{M_i}{\sigma_i} \quad (5.7)$$

Appliquons maintenant ces étapes aux données brutes de positions données respectivement par le Scalable-SPIDAR et le système de tracking optique ARTtrack. Mais, pour plus de clarté nous utiliserons une écriture matricielle :

$$\begin{cases} N_{\text{SPIDAR}} = (P_{\text{Scalable-SPIDAR}} - \Gamma_{\text{Scalable-SPIDAR}}) \cdot \sum_{\text{Scalable-SPIDAR}} \\ N_{\text{OTS}} = (P_{\text{ARTtrack1}} - \Gamma_{\text{ARTtrack1}}) \cdot \sum_{\text{ARTtrack1}} \end{cases} \quad (5.8)$$

$$\Gamma_{\text{Scalable-SPIDAR}} = \Upsilon_{\text{Scalable-SPIDAR}} \cdot J \text{ et } \sum_{\text{Scalable-SPIDAR}} = \sigma_{\text{Scalable-SPIDAR}} \cdot J \quad (5.9)$$

$$\Gamma_{\text{ARTtrack1}} = \Upsilon_{\text{ARTtrack1}} \cdot J \text{ et } \sum_{\text{ARTtrack1}} = \sigma_{\text{ARTtrack1}} \cdot J \quad (5.10)$$

$\Upsilon_{\text{Scalable-SPIDAR}}$: Matrice des moyennes des composantes de la matrice des positions récoltées avec le Scalable-SPIDAR.

$\Upsilon_{\text{ARTtrack1}}$: Matrice des moyennes des composantes de la matrice des positions récoltées avec le système de tracking optique ARTtrack1.

$P_{\text{Scalable-SPIDAR}}$: Positions brutes données par le Scalable-SPIDAR

$P_{\text{ARTtrack1}}$: Positions brutes données par le système de tracking optique ARTtrack1

J : Matrice identité

N_{SPIDAR} : Positions normalisées du Scalable-SPIDAR

$N_{\text{ARTtrack1}}$: Positions normalisées du système ARTtrack1

Une fois les positions brutes sont normalisées nous entamons la phase d'apprentissage. Pour cela, nous avons utilisé les positions de Scalable-SPIDAR comme entrées du réseau de neurones et les positions de tracking optique comme sorties. De plus, les données récoltées avec notre protocole sont découpées aléatoirement en trois sous ensembles (figure 5.9) :

- Un premier sous-ensemble (60% de données) pour l'apprentissage proprement dit.
- Un deuxième sous-ensemble (20% de données) utile pour la validation de l'apprentissage et s'assurer qu'il y a pas un phénomène de sur-apprentissage.
- Un troisième et dernier sous ensemble de test (20% de données) qui sert à évaluer les performances de généralisation du réseau, c'est à dire lors de la présentation des nouvelles données qui n'ont pas servi à l'apprentissage.

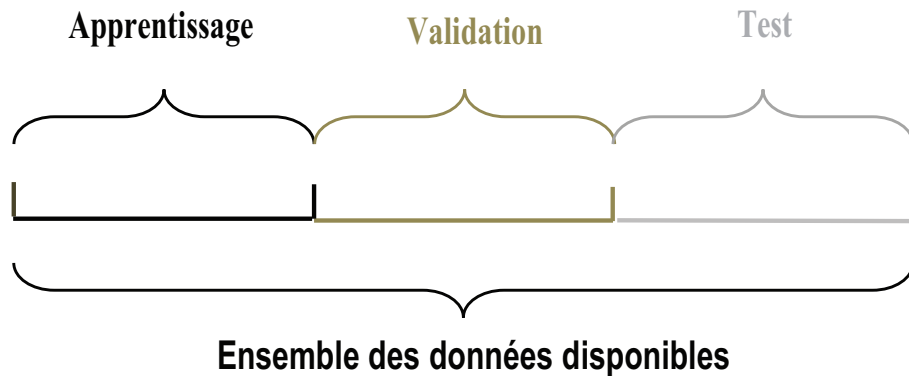


FIGURE 5.9 – Décomposition des données récoltées en trois sous ensembles : apprentissage, validation et test

5.3.1.2 L'approche expérimentale

Comme nous l'avons vu dans le chapitre 3, les réseaux de neurones se distinguent par leur capacité d'approximation parcimonieuse.

Le type de réseau que nous utilisons dans notre travail est un réseau non bouclé à couches, dont les neurones cachés ont une fonction d'activation sigmoïde et qui est également appelé "Perceptron multicouche" (ou MLP). Ce choix est justifié par ses qualités d'approximateur universel et sa faisabilité comme outil de régression.

5.3.1.2.1 Configuration et optimisation des caractéristiques du réseau de neurones

Pour déterminer la meilleure configuration du réseau de neurones, nous employons l'approche constructive pour optimiser le réseau de neurones pendant l'étape d'apprentissage. Le choix du nombre de couche cachées et du nombre de neurones nécessaires au problème de l'étude n'est pas déterminable théoriquement. En effet, il n'existe aucune théorie valable spécifiant la configuration optimale qu'il faut donner à un réseau de neurones pour permettre l'estimation de la fonction de régression de manière précise et rapide. La propriété de généralisation du réseau est également un paramètre à ne pas prendre à la légère. En effet, si le réseau apprend trop parfaitement les données, la généralisation peut être médiocre et de ce fait, il y aurait une mauvaise estimation d'une nouvelle donnée qui apparaît à l'entrée du réseau.

Comme nous l'avons expliqué dans le chapitre 3, notre choix s'est porté à un réseau de type Perceptron multicouche à une seule couche cachée avec calcul de poids qui s'effectue par rétropropagation du gradient de l'erreur. La configuration de base comporte une seule couche cachée dont la détermination du nombre de neurones approprié fera l'objet d'une optimisation, et une couche de sortie composée de trois neurones. La couche cachée est constituée de fonctions d'activation non-linéaire de type sigmoïde tangente tandis que la couche de sortie comporte des fonctions d'activations linéaires de type rampe (figure 5.10).

Nous utilisons l'algorithme de Levenberg-Marquardt pour le calcul des poids suivant la cible présentée en sortie du réseau car il est le plus rapide en termes de convergence. L'algorithme d'apprentissage est la descente du gradient à taux d'apprentissage fixe de 0.001 avec un terme inertiel qui agit comme filtre passe-bas pour éliminer les oscillations sur la surface d'erreur et augmenter un peu la vitesse de convergence.

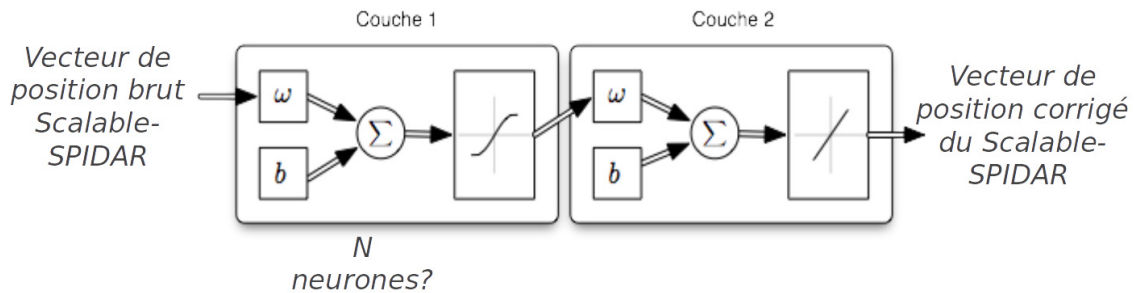


FIGURE 5.10 – Architecture du réseau de neurones de base utilisé pour la calibration du Scalable-SPIDAR

Une fois le nombre de couches cachées est fixé, il reste à déterminer le nombre de neurones N requis dans cette couche cachée. Cependant, nous ne pouvons pas déterminer ce nombre au hasard et il est très problématique de le trouver car jusqu'à maintenant il n'y a pas de théorie stipulant le nombre de neurones nécessaire par couche permettant une configuration optimale. En effet, si la couche cachée comprend un trop grand nombre de neurones, la fonction à approximer peut être surévaluée. Autrement dit, tous les points appartiennent à la fonction évaluée, à l'exception de ceux entre les points, nous obtenons des valeurs qui n'appartiennent plus à la fonction. La fonction subit alors des oscillations entre ces points, dû au grand nombre de degrés de liberté du réseau. Par conséquent, il ne faut pas perdre de vue que la capacité de généralisation du réseau est une caractéristique très importante dans l'approximation de la fonction de régression. Donc il ne faut pas que le réseau soit surévalué pour obtenir une bonne généralisation.

Une approche intéressante sur la recherche de nombre de neurones dans un couche cachée apparaît dans l'article de [KINDRATENKO et SHERMAN \[2005\]](#) concernant la calibration des systèmes de tracking électromagnétiques. Les auteurs sont arrivés à déterminer le nombre de neurones N dans la couche cachée en utilisant une technique constructive. A cet effet, différentes architectures seront expérimentées en variant le nombre de neurones dans la couche cachée de 3 à 21 en fonction de l'erreur moyenne absolue. La figure 5.11 permet d'identifier la meilleure configuration parmi celles expérimentées, la meilleur est celle possédant 18 neurones. Nous obtenons une erreur absolue moyenne de l'ordre de $16mm$.

Les différentes architectures ont été entraînées par simulation à plusieurs reprises chacune, en variant le nombre de neurones dans la couche cachée, et avec des poids et des biais aléatoires : les poids obtenus après la phase d'apprentissage ont été ensuite utilisés pour tester la fiabilité du réseau, caractérisée par l'obtention d'une erreur moyenne quadratique de test minimale.

Il se peut que les résultats relatifs à la configuration du réseau de neurones ne soient pas optimaux mais à la vue des résultats obtenus et avec la vitesse à laquelle l'apprentissage s'est déroulée, nous pouvons aboutir, sans trop nous tromper, à une bonne optimisation du réseau.

La figure 5.12 représente l'évolution de la fonction de coût du Scalable-SPIDAR lors de l'étape de l'apprentissage, de validation et de test. Nous distinguons clairement la stabilisation de l'erreur d'apprentissage après quelques itérations.

La fonction coût est l'erreur quadratique moyenne $F = mse = \frac{1}{N} \sum_{i=1}^N e_i^2 = \frac{1}{N} \sum_{i=1}^N (t_i - a_i)^2$ soit la moyenne de la somme du carré des erreurs entre la sortie du réseau a_i et la sortie désirée t_i sur toute la collection de données.

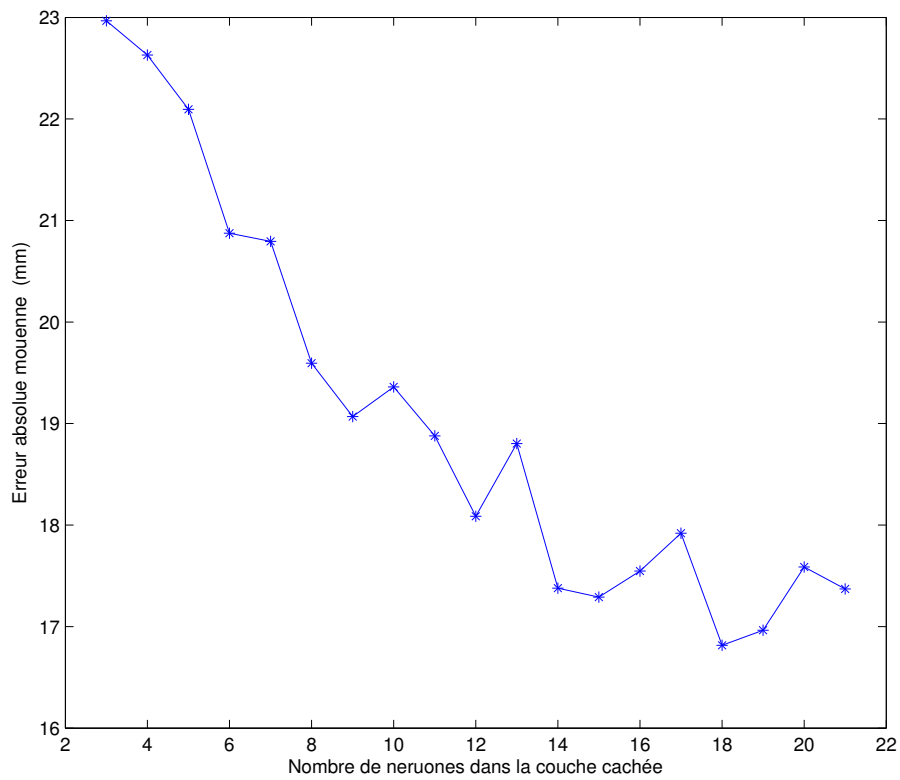


FIGURE 5.11 – Erreur absolue moyenne en fonction du nombre de neurones dans la couche cachée

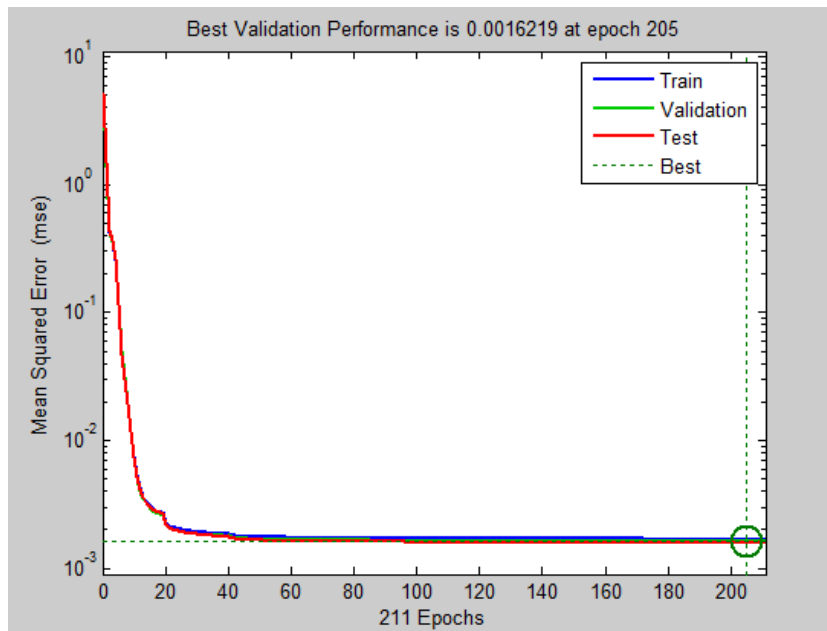


FIGURE 5.12 – Performance du réseau de neurones employé pour la calibration

5.3.1.2.2) Performances du réseau de neurones en apprentissage

Afin de mettre en évidence les performances du réseau de neurones développé, nous avons choisi de nous appuyer sur un ensemble de graphiques et des valeurs statistiques représentatives c'est-à-dire :

- ✧ un graphique en 3D représentant la distribution spatiale de l'erreur absolue à chaque point de référence (x_r, y_r, z_r) . Notons que la couleur de l'erreur est proportionnelle à son amplitude.
- ✧ L'histogramme des erreurs absolues de position
- ✧ L'erreur moyenne absolue de position
- ✧ L'écart type sur les erreurs absolues
- ✧ L'erreur absolue maximale

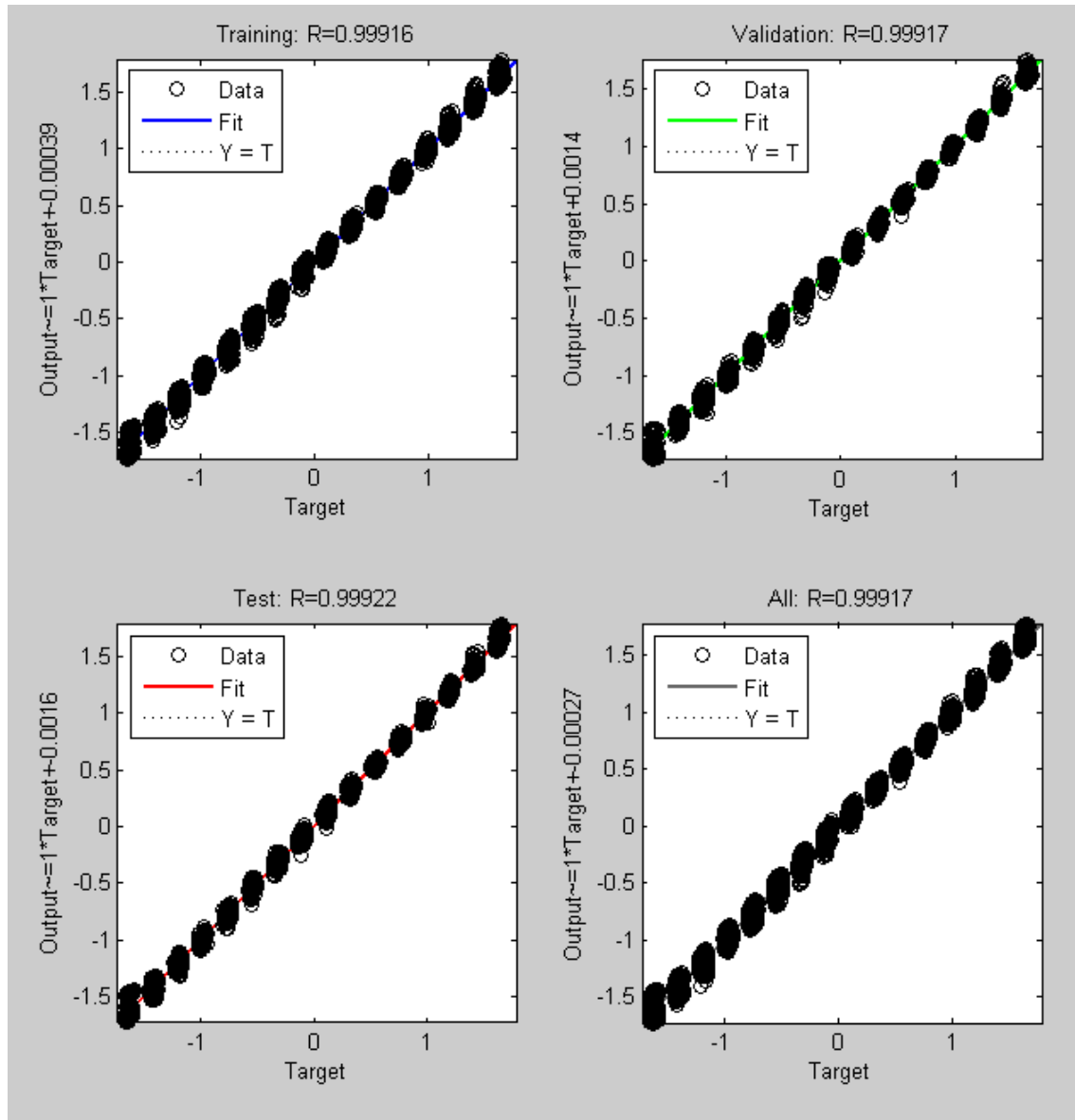


FIGURE 5.13 – Régression en fonction des sous ensembles des données utilisées

La figure 5.13 représente la corrélation entre les sorties désirées ou encore les positions données par le système de tracking optique ARTrack1 et les sorties produites par le réseau pour les trois sous ensembles de données : apprentissage, validation et test. Plus la fonction modélisée s'apparente à une droite, plus le réseau de neurones est performant. Pour chacun de sous-ensembles, la boîte à outils de Matlab fournit l'équation de la droite de régression linéaire de meilleur ajustement entre les sorties du réseau et les données ciblées du graphique concerné. L'équation est de la forme : $Sortie = R * cible + constante$, où R désigne le coefficient de corrélation.

Le coefficient de corrélation dénote aussi le coefficient directeur de la droite de régression linéaire compris entre -1 et 1. Plus le coefficient de corrélation est proche de 1 et plus les données sont fortement corrélées. Les valeurs obtenues nous informent sur la qualité de la correction apportée par les réseaux de neurones. Ainsi, avec un coefficient de corrélation cumulé de 0.99917, il est clair que notre réseau de neurones est suffisamment efficace et permet d'obtenir des données en sortie qui sont fortement corrélées avec les données désirées.

La figure 5.14 représente l'erreur absolue de position du Scalable-SPIDAR avant et après calibration. Il est clair qu'avant calibration, l'erreur absolue de position retournée par le Scalable-SPIDAR est très sensible à la distance du centre. Néanmoins, cette erreur peut aussi être faible même en étant loin du centre. C'est pour cette raison que le graphique de gauche ne présente pas une courbe, mais plutôt une surface parallélogrammique, soulignant l'aspect non linéaire de la distribution de l'erreur absolue dans notre espace de travail. Quant au graphique de droite, il souligne l'intérêt du réseau de neurones. En effet, nous pouvons très bien remarquer une nette amélioration de la précision (le nuage de points est complètement écrasé). L'erreur absolue de position par rapport au centre est répartie de manière beaucoup plus homogène après calibration, d'où une bonne homogénéisation de la précision.

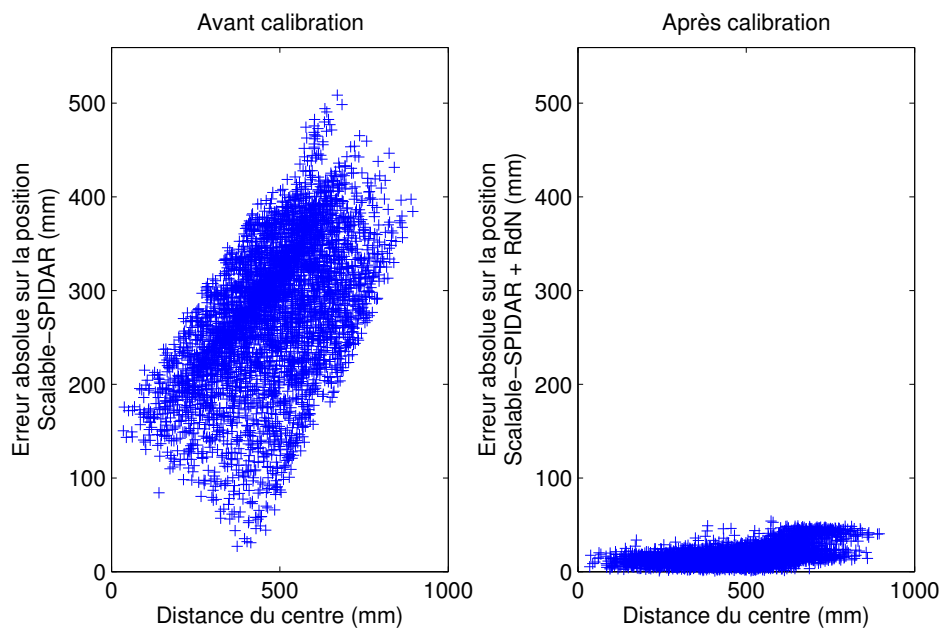


FIGURE 5.14 – Erreur absolue de position du Scalable-SPIDAR par rapport à la distance du centre avant et après calibration par réseau de neurones

Cette constatation effectuée, considérons maintenant la représentation spatiale de la distribution de l'erreur absolue de position (figure 5.15). Nous pouvons constater que le réseau de neurones a clairement amélioré la précision de position du Scalable-SPIDAR. La quasi-totalité de l'espace de travail comporte des erreurs inférieures à 60mm. Plus nous approchons du centre, plus la précision de position augmente. En effet, nous disposons d'une plage d'erreurs diminuée de 10 par rapport à la première distribution relevée (figure 5.2) et donc d'une gamme de couleurs différentes.

Afin d'obtenir une meilleure idée sur l'étendue des erreurs, nous avons tracé l'histogramme des erreurs absolues de position du Scalable-SPIDAR par rapport à celle du système de tracking optique ARTtrack1.

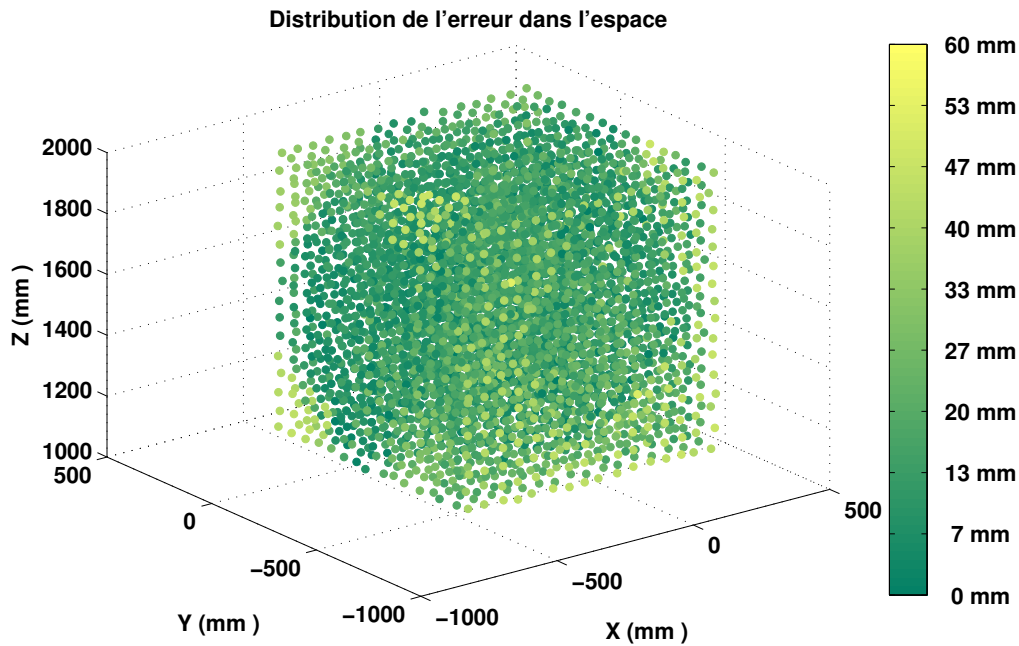


FIGURE 5.15 – Représentation spatiale de l’erreur absolue après calibration par le réseau de neurones

La figure 5.16 ainsi que le tableau 5.2 représentent respectivement l’histogramme des erreurs absolues de position du Scalable-SPIDAR avant et après calibration par le réseau de neurones ainsi que les valeurs statistiques caractéristiques de ces erreurs.

TABLEAU 5.2 – Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après calibration par le réseau de neurones

Erreur absolue	Avant calibration (Brut)	Après calibration avec RDN
Erreur moyenne (mm)	274.82	16.96
Ecart type (mm)	78.12	9.72
Erreur maximale (mm)	508.56	54.11

L’histogramme étalé témoigne d’une mauvaise précision de position du Scalable-SPIDAR : c’est le cas avant calibration. Contrairement, un histogramme qui est décalé vers le côté gauche présente une bonne précision de position du Scalable-SPIDAR : c’est le cas après calibration par le réseau de neurones. Ces résultats sont appuyés par l’observation des valeurs statistiques caractéristiques des erreurs absolues. A la lecture de tableau, nous constatons que l’ordre de grandeur de la moyenne des erreurs absolues a diminué d’une manière très significative d’environ 275mm pour le Scalable-SPIDAR avant calibration (BRUT dans le tableau) à 16.9mm pour le Scalable-SPIDAR calibré par le réseau de neurones (RDN dans le tableau). L’écart-type, reflétant la dispersion des erreurs absolues de position, quant à lui, a été divisé par 9, passant d’environ 78.1mm à 9.7mm pour le Scalable-SPIDAR calibré.

5.3.1.2.3) Performances du réseau de neurones en généralisation

Pour bien évaluer la pertinence de notre réseau de neurones, il ne suffit pas juste de le tester avec les données qui ont été utilisées lors de l’étape de l’apprentissage. Il faut égale-

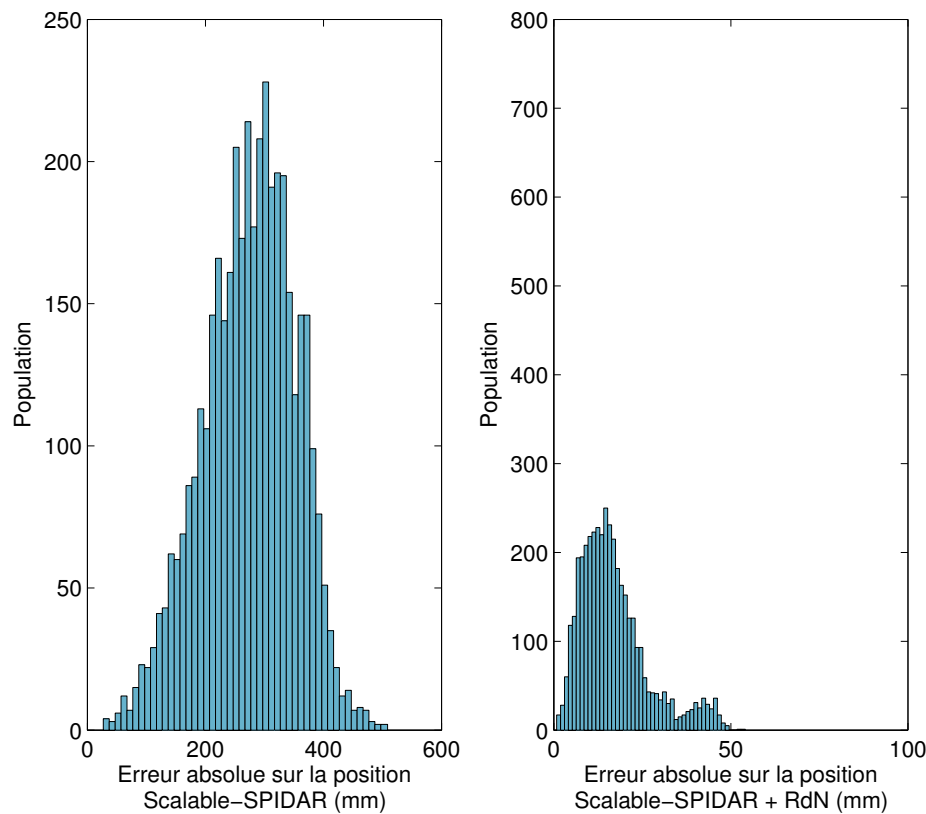


FIGURE 5.16 – Erreur absolue de position du Scalable-SPIDAR avant et après calibration à l'aide du réseau de neurones

ment le solliciter avec des données auxquelles il n'a jamais été confronté. Nous appelons cette procédure la "généralisation".

Ces données sont récoltées lors d'une campagne de 1024 mesures, que nous nommons "ensemble de données en généralisation", durant laquelle nous suivons des trajectoires aléatoires avec l'effecteur du Scalable-SPIDAR, en récupérant à la fois les positions données par le Scalable-SPIDAR et celles renvoyées par le système de tracking optique ARTtrack1.

Nous pouvons constater que notre réseau de neurones garde de bonnes performances avec des données en généralisation. En effet, à la lecture du tableau 5.3 l'erreur absolue moyenne a augmentée pour atteindre 24mm . De même, la dispersion statistique a aussi passé de 9.7 à 14.5mm .

TABEAU 5.3 – Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après généralisation

Erreur absolue	Brut	RDN
Erreur moyenne (mm)	273.48	24.04
Ecart type (mm)	80.92	14.58
Erreur maximale (mm)	518.80	72.64

A la lumière de ces graphiques (figure 5.17), nous pouvons conclure que notre réseau de neurones est robuste aux données qui lui sont présentées, dès lors qu'elles résident à l'intérieur de l'espace de travail du Scalable-SPIDAR. Ceci n'est pas surprenant du fait de

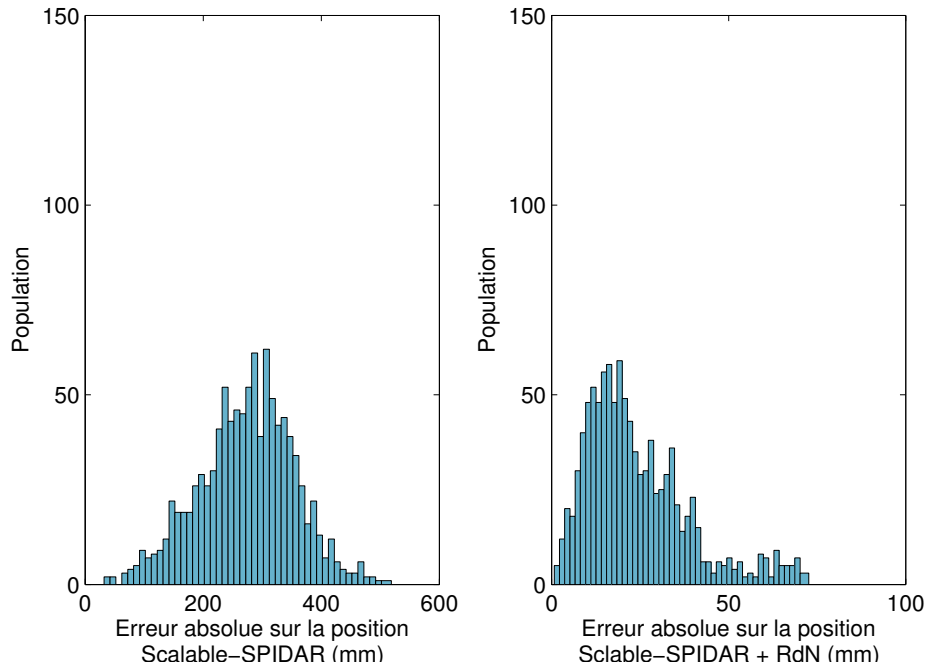


FIGURE 5.17 – Erreur absolue de position du Scalable-SPIDAR avant et après généralisation

la qualité de données acquises par le nouveau protocole. En effet, tel qu'il a été conçu, notre protocole permet une description représentative et complète de l'espace de travail à calibrer. De plus, la quantité et la granularité de données de celles-ci dans l'espace permettent un apprentissage capable de couvrir la quasi-totalité de l'espace de travail du Scalable-SPIDAR, garantissant ainsi au réseau de neurones la capacité d'obtenir une très bonne interpolation. Néanmoins, comme nous l'avons expliqué dans le chapitre 3, l'inconvénient des réseaux de neurones est que nous ne pouvons pas être sûres que nous sommes tombés sur un minimum local et donc, nous ne pouvons en aucun cas garantir que nos résultats soient optimaux. C'est pour cette raison, nous avons opté de tester la méthode de régression à vecteurs de support.

5.3.2 Calibration du Scalable-SPIDAR à l'aide des SVR

Pour vérifier les performances de notre réseau de neurones, nous avons utilisé les SVR qui sont l'application des machines à vecteurs de support au problème de régression. En effet, les SVR sont des approximateurs universels parcimonieux. Une particularité intéressante est l'unicité du minimum global contrairement aux réseaux de neurones qui souffrent de l'existence de plusieurs minima locaux.

A la fin du chapitre 3, nous avons décrit la méthodologie employée lors de la phase expérimentale pour construire un modèle à base des machines à vecteurs de support. Le choix du noyau constitue une étape fondamentale pour la modélisation boîte noire. Notre choix s'est porté sur un noyau gaussien $k(\mathbf{x}, \mathbf{x}_i) = \exp\left(\frac{-\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}\right)$ où σ désigne la largeur de la gaussienne.

Pour déterminer de manière simple et efficace les différents paramètres des SVR, nous utilisons la méthode de sélection "recherche de grille" ou "Grid Search". Le triplet de paramètres qui nous offre les meilleures performances est $\{\epsilon = 0.01, \gamma = 0.0125, C = 3\}$

5.3.2.1 Performances des SVR en apprentissage

Comme pour les réseaux de neurones, nous représentons la répartition de l'erreur absolue en fonction de la position de l'effecteur par rapport au centre avant et après calibration (figure 5.18). Il est clair que la correction de la position du Scalable-SPIDAR par les SVR est efficace. Nous constatons, une nette diminution des erreurs absolues de position. Ces résultats semblent être en accord avec ceux obtenus avec le réseau de neurones. Il faut cependant noter une dispersion statistique presque identique à celle obtenue avec le réseau de neurones.

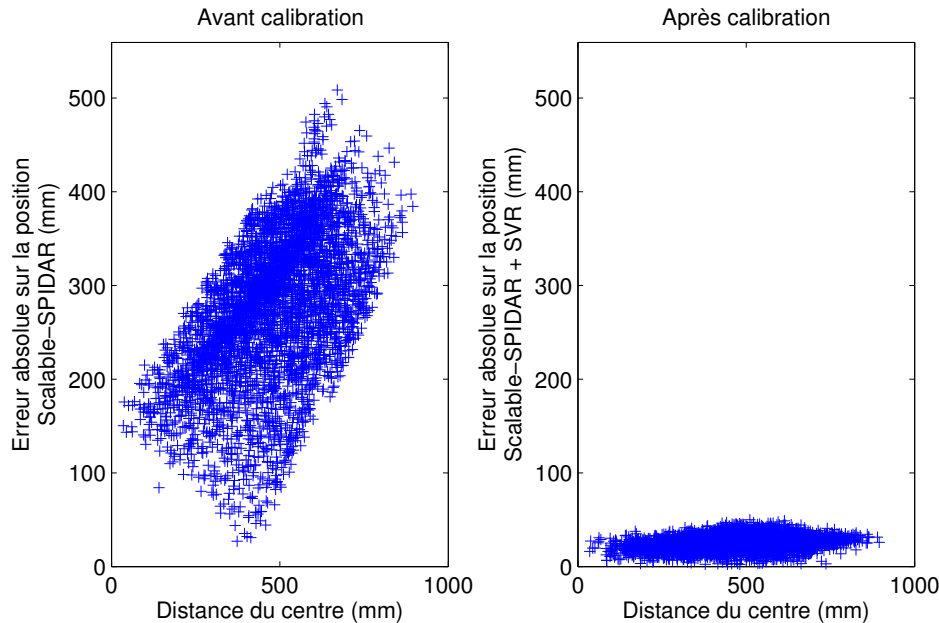


FIGURE 5.18 – Erreur absolue de position du Scalable-SPIDAR par rapport à la distance du centre avant et après calibration par SVR

La figure 5.19 met en évidence les bonnes performances de la calibration par les SVRs. En effet, la majorité des erreurs absolues supérieures à 50mm sont situées dans les coins de l'espace de travail, alors qu'à l'intérieur de celui-ci, l'erreur est inférieure à 20mm . Nous remarquons également que ce graphique est très identique à celui obtenu après calibration par le réseau de neurones.

La figure 5.20 et le tableau 5.4 montrent l'intérêt de la calibration par les SVRs. L'erreur absolue moyenne de position a considérablement diminué de 274.82mm à 15.60mm et par conséquent une nette augmentation de la précision moyenne. L'écart-type a également très fortement diminué passant ainsi de 78.12mm à 9.35mm , ce qui témoigne d'une forte diminution de la dispersion statistique.

TABEAU 5.4 – Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après calibration à l'aide des SVR

Erreur absolue	Avant calibration	Après calibration avec SVR
Erreur moyenne (mm)	274.82	15.60
Ecart type (mm)	78.12	9.35
Erreur maximale (mm)	508.56	58.63

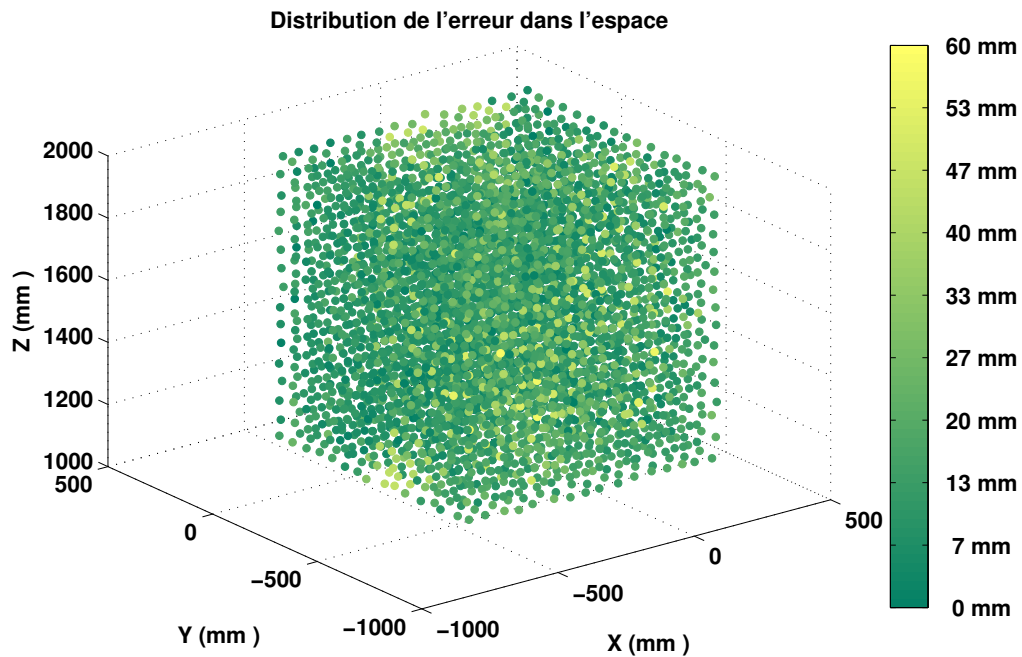


FIGURE 5.19 – Représentation spatiale de l'erreur absolue après calibration par les SVR

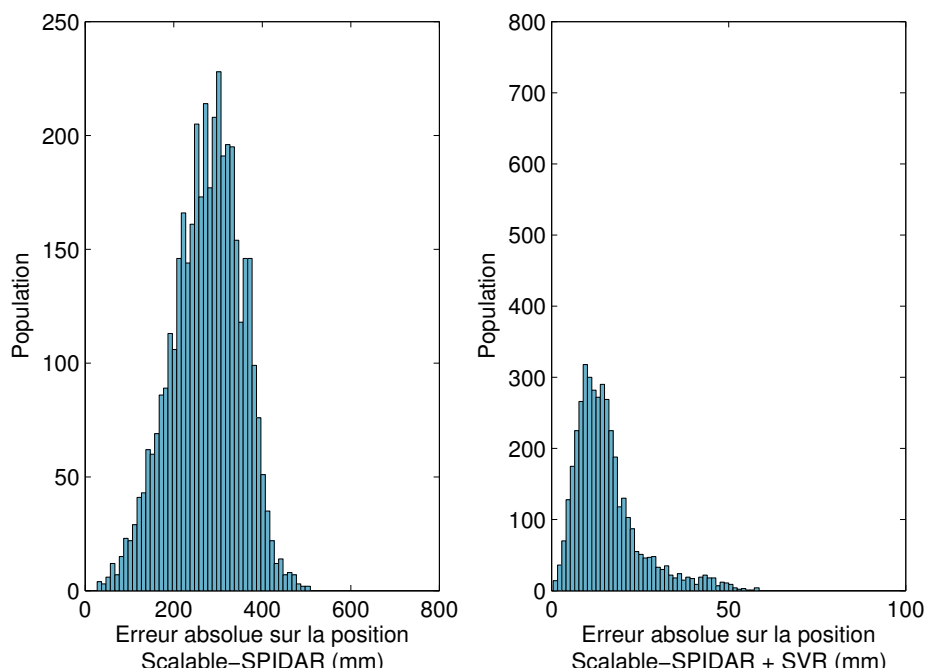


FIGURE 5.20 – Erreur absolue de position du Scalable-SPIDAR avant et après calibration à l'aide des SVR

5.3.2.2 Performances des SVR en généralisation

Pour bien évaluer la capacité de généralisation des SVR, nous présentons de nouvelles données qui n'ont pas servi à l'apprentissage. Ces données ont été récoltées lors d'une campagne de mesure pendant laquelle nous avons emprunté des trajectoires aléatoires avec l'effecteur du Scalable-SPIDAR.

Nous pouvons constater que les SVR gardent de très bonnes performances avec des

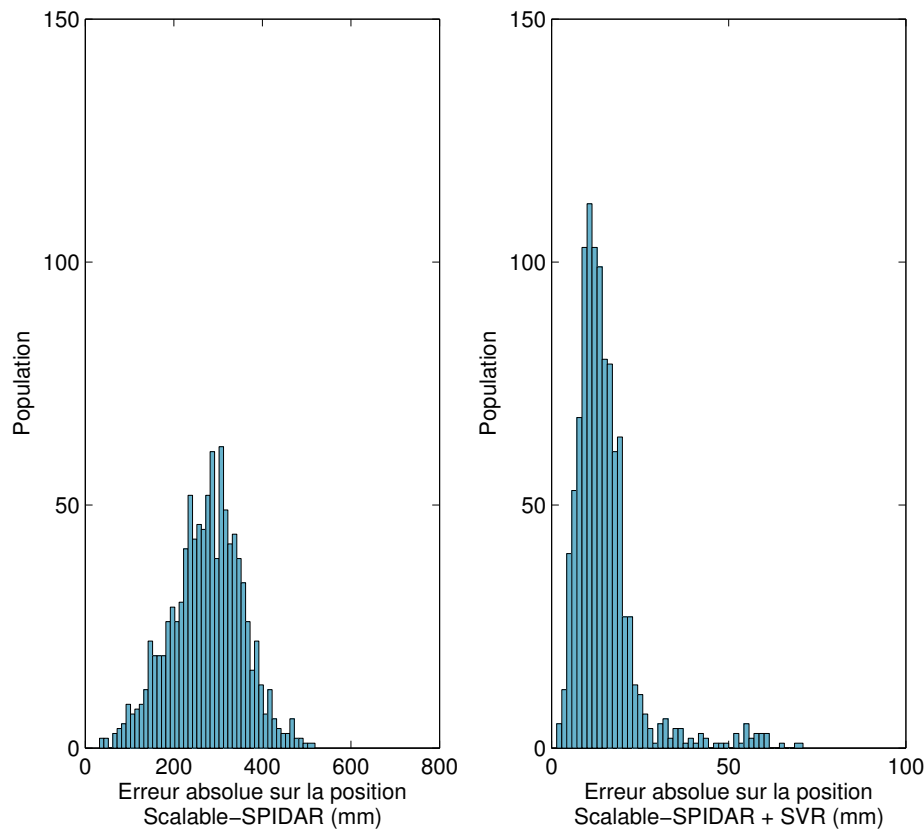


FIGURE 5.21 – Erreur absolue de position du Scalable-SPIDAR avant et après généralisation à l’aide des SVR

TABLEAU 5.5 – Valeurs caractéristiques des erreurs absolues de position du Scalable-SPIDAR après généralisation à l’aide des SVR

Erreur absolue	Brut	SVR
Erreur moyenne (mm)	273.48	14.90
Ecart type (mm)	80.92	9.1
Erreur maximale (mm)	518.80	70.97

données en généralisation aussi bonnes que celles fournis par les réseaux de neurones. En effet, l’erreur absolue moyenne ainsi que l’écart type sont presque identiques aux résultats obtenus en apprentissage. L’erreur absolue maximale a grimpé pour atteindre $70mm$.

A la lumière de ces résultats nous pouvons conclure que les SVR conviennent aux données qui lui sont présentées dès lors qu’elle réside à l’intérieur de l’espace de travail du Scalable-SPIDAR.

5.3.3 Mise en balance des méthodes de calibration

Comme le montre la figure 5.22 et le tableau 5.6, la calibration par SVR offre des performances meilleures en généralisation que les réseaux de neurones, avec une erreur absolue moyenne de $14.9mm$. En seconde place vient la calibration par les réseaux de neurones avec $24.04mm$ d’erreur moyenne. En terme, de dispersion statistique, le classement des méthodes de calibration est le même. Ainsi, les SVR est la méthode souffrant le moins

de dispersion statistique avec une valeur de 9.11 mm , puis vient l'approche par les réseaux de neurones avec une dispersion d'environ 14.58 mm . Les résultats de la régression à vecteurs de support permettent de garantir la bonne convergence du réseau de neurones vers un minimum global et de s'assurer que le réseau sélectionné est bien optimisé.

TABLEAU 5.6 – Comparaison des valeurs caractéristiques des erreurs absolues de deux méthodes en généralisation

Erreur absolue	Brut	SVR	RdN
Erreur moyenne (mm)	273.48	14.9	24.04
Ecart type (mm)	80.92	9.11	14.58
Erreur maximale (mm)	518.8	70.97	72.64

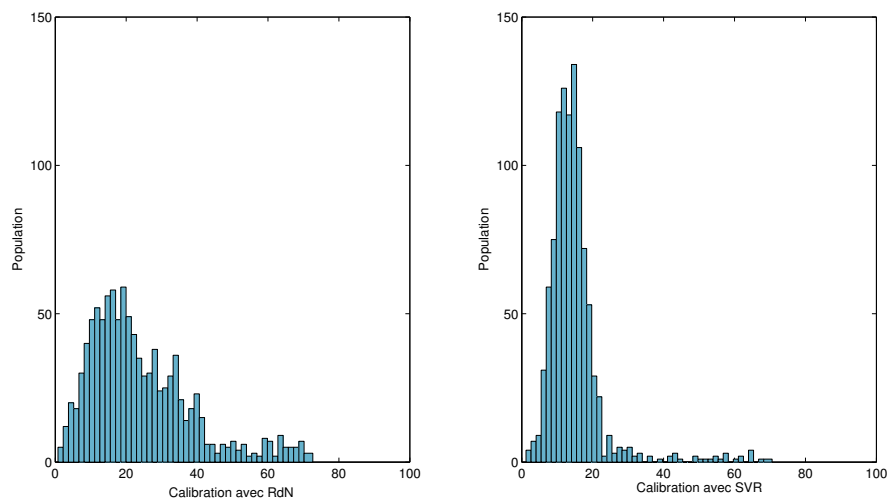


FIGURE 5.22 – Comparaison entre les performances de deux méthodes de calibration réseaux de neurones et SVR en généralisation

5.4 MISE EN PRATIQUE DE LA PROCÉDURE DE CALIBRATION DU SCALABLE-SPIDAR

Après avoir choisi la technique de calibration, nous décrivons la mise en pratique de notre procédure de calibration notamment par la conception d'un module de collecte et de correction des données permettant la calibration du Scalable-SPIDAR en temps réel. Cette unité se place entre le logiciel de la réalité virtuelle et les interfaces matérielles de réalité virtuelle, soit le Scalable-SPIDAR de notre cas.

Nous avons conçu une application web afin d'offrir une interface graphique à l'utilisateur pour qu'il puisse choisir les interfaces matérielles qu'il désire employer et les traitements à appliquer. A l'issue de cette phase préliminaire, un fichier XML est généré. A partir de l'interprétation de ce fichier, cette unité initialise les communications avec les interfaces sélectionnées par l'utilisateur. Une fois cette étape effectuée, elle exécute les traitements mathématiques à appliquer sur les données. Ensuite, elle mettra à disposition les données obtenues sur le réseau. D'un point de vue pratique, cette unité agit comme une

interface matérielle virtuelle ou un ensemble d'interfaces matérielles virtuelles avec lesquelles l'application de réalité virtuelle dialogue. D'autre part, une telle conception nous garantit que la gestion matérielle ainsi que les traitements sur les données, qui souvent utilisent beaucoup de ressources, ne viendra pas perturber les performances de l'application de réalité virtuelle.

Dans ce qui suit, nous présentons brièvement les différents modules constituant l'unité de collecte et de correction de données (figure 5.23).

Gestionnaire des interfaces matérielles Ce module permet de dialoguer avec les différentes interfaces à l'aide de plusieurs moyens : l'utilisation d'APIs spécifiques ou d'autres clients plus appropriés.

Le collecteur de données Ce module permet de filtrer les données provenant du gestionnaire des interfaces matérielles afin de ne laisser passer que les données utiles et valides.

Correspondance Ce module permet de ramener les données, venant des différentes interfaces matérielles et exprimées dans des systèmes de coordonnées différents, dans un autre espace de représentation commun. Pour cela, nous utiliserons la bibliothèque mathématique fournie par MATLAB.

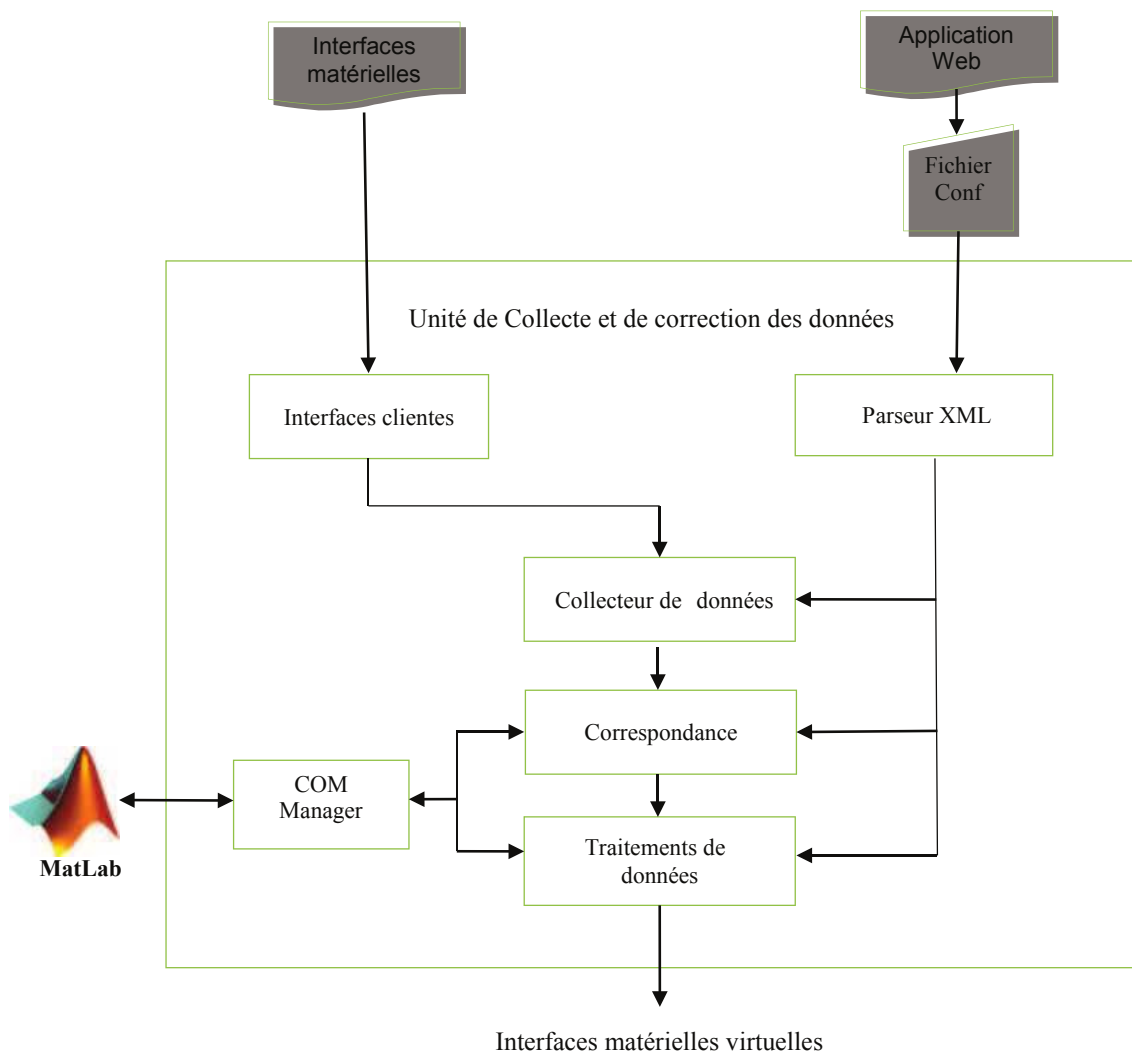


FIGURE 5.23 – L'unité de collecte et de correction de données

Traitements des données Dans ce module sont exécutés les traitements sélectionnés au préalable dans le fichier XML, sur les données. Pour la calibration, nous utiliserons les méthodes de calibration déjà développées sous MATLAB.

Le parseur XML Ce module découpe le fichier XML en trois sections (initialisation des interfaces matérielles, description des interfaces matérielles virtuelles et description des événements). Celles-ci sont transformées ensuite en processus parallèles qui s'appliqueront sur les données.

Manager com et MATLAB Afin d'offrir à l'utilisateur de l'unité de collecte et de correction de données une large panoplie de fonctions mathématiques, nous y avons intégré un manager COM, dans le but de faciliter l'accès à tous les outils mathématiques de MATLAB.

La figure 5.24 décrit un schéma simplifié décrivant les différentes étapes du processus de traitement de données dans l'unité de collecte et de correction de données. La boucle de traitement consiste à :

- D'abord, récupérer les données des différentes interfaces matérielles,
- Ensuite, appliquer les traitements sur les données et créer par conséquent des interfaces virtuelles,
- Enfin, mettre sur le réseau les données calculées venant des différentes interfaces virtuelles.

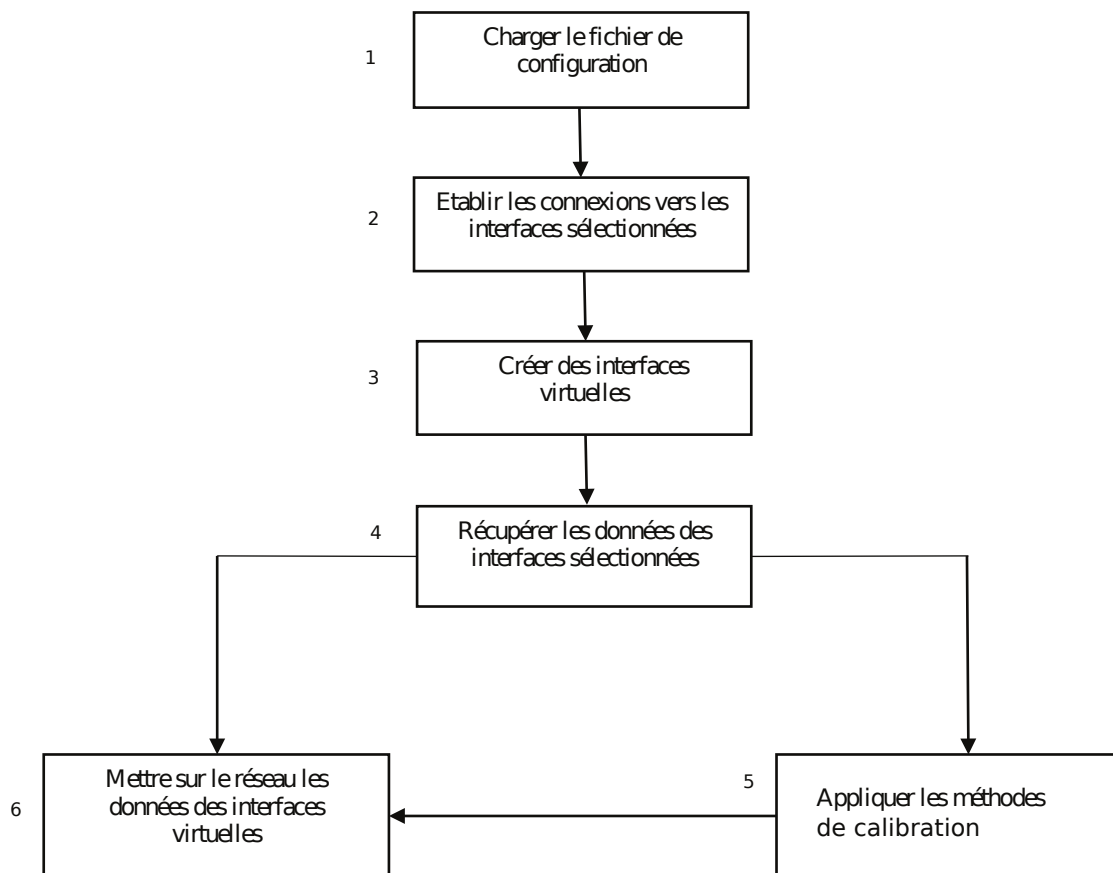


FIGURE 5.24 – Boucle de traitement de données

5.5 APPLICATIONS PRATIQUES

Une fois la meilleure méthode choisie, nous concevons deux applications de réalité virtuelle pour mettre en relief l'intérêt des méthodes de traitement de données.

La première application consiste à mettre en place un simulateur virtuel de notre Scalable-SPIDAR. Ce simulateur permet de retracer les trajectoires effectuées par l'effecteur après calibration et les comparer à des trajectoires de référence. Ainsi, nous créons un modèle virtuel du Scalable-SPIDAR permettant d'évaluer les qualités de méthodes de calibration directement sur l'espace de travail concerné.

La deuxième application s'inscrit dans un cadre plus large : la téléopération. Le but de cette application est de piloter une station robotisée distante depuis notre plateforme de réalité virtuelle.

Dans la prochaine section, nous revenons sur les phases importantes dans la conception de ces applications de réalité virtuelle.

5.5.1 Modélisation 3D

La première étape de notre travail est d'essayer de construire des représentations 3D des objets géométriques étudiés. Afin d'être plus proche de la réalité, nous avons choisi de nous baser sur les éléments caractéristiques de chacun des objets.

5.5.1.1 Contraintes architecturales

La modélisation réaliste de la scène de notre application nécessite le respect de dimensions réelles de l'interface à retour d'effort le Scalable-SPIDAR ainsi que de la station robotisée ACRS (Automatic Storage and Retrieval System). Le Scalable-SPIDAR est constitué d'une structure cubique dont l'arrête est de 3m. Le rayon de l'effecteur de forme sphérique est de l'ordre de 3 centimètres. Les principales dimensions de la station ACRS sont illustrées dans la figure 5.25.

5.5.1.2 Techniques de modélisation

Deux techniques de modélisation volumiques sont utilisées pour créer les modèles 3D de la station ACRS et du Scalable-SPIDAR : la représentation par arbre de construction CSG (constructive Solid Geometry) et les modèles polyédriques (maillage).

La première technique repose sur le fait qu'il est possible de construire des objets complexes comme une composition d'objets simples appelés "primitives 3D standards". A titre d'exemples les sphères, les cubes, les cylindres, etc. Nous pouvons donc créer des objets plus complexes en employant des opérateurs booléens tels que l'union ou la soustraction sur ses primitives. De plus, nous pouvons utiliser des transformations géométriques telles que la translation, la rotation ou l'homothétie. La figure 5.26 illustre un exemple de la représentation CSG d'une sphère A et d'un cône B en utilisant différents types des opérateurs booléens.

Un objet CSG se présente sous la forme d'un arbre dont les nœuds internes sont les opérateurs booléens et dont les feuilles sont les primitives qui forment l'objet 3D ayant subi un certain nombre de transformations.

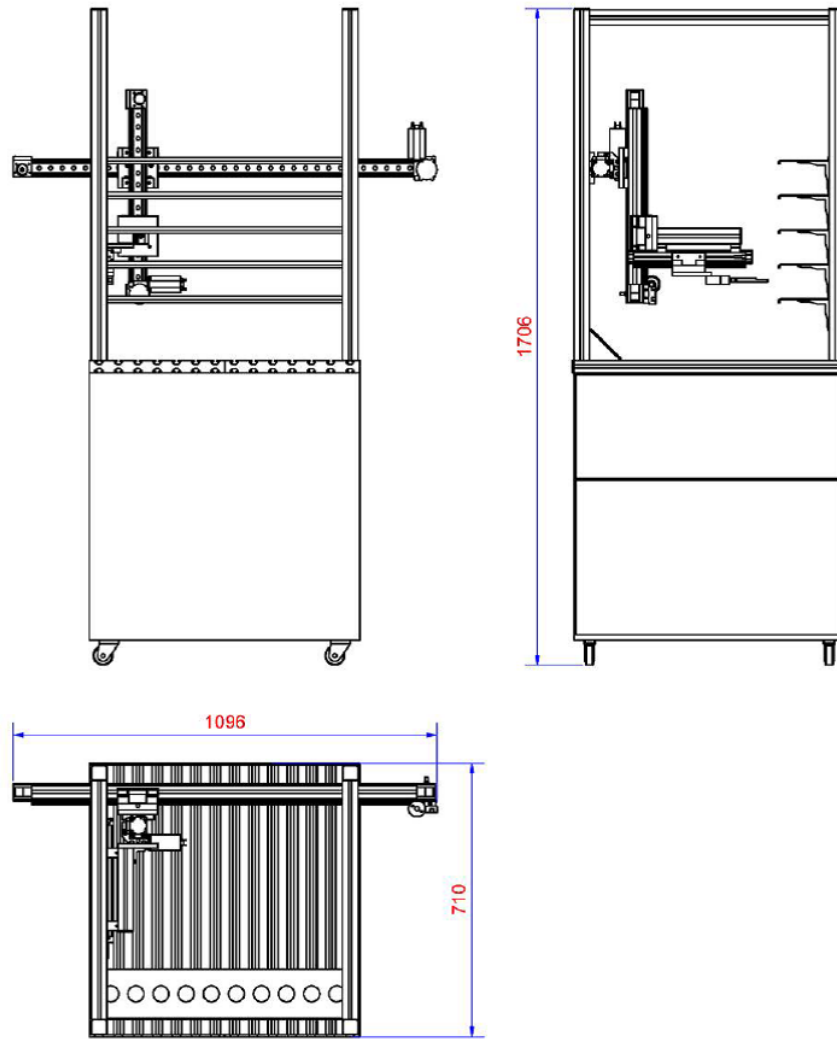


FIGURE 5.25 – Dimensions de la station robotisée ACRS

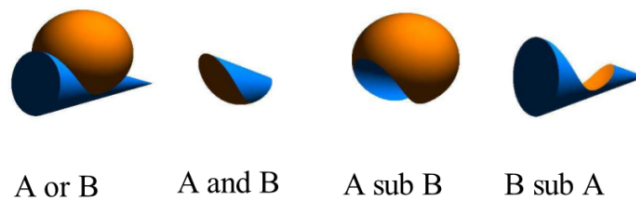


FIGURE 5.26 – Exemple des opérateurs booléens.

La deuxième technique est, quant-à elle, une représentation discrète d'une forme ou de la surface d'un objet. Cette architecture décompose la surface d'un objet en éléments finis appelés "facettes". Une facette est de forme polygonale et créée à partir de plusieurs arrêtes qui sont de même formées par des sommets. Les facettes peuvent-être quadrangulaires mais elles sont d'une manière générale choisies comme étant des triangles.

Cette représentation permet de manipuler chaque facette, arrête et sommet de la surface d'un objet 3D d'une manière indépendante (figure 5.27).

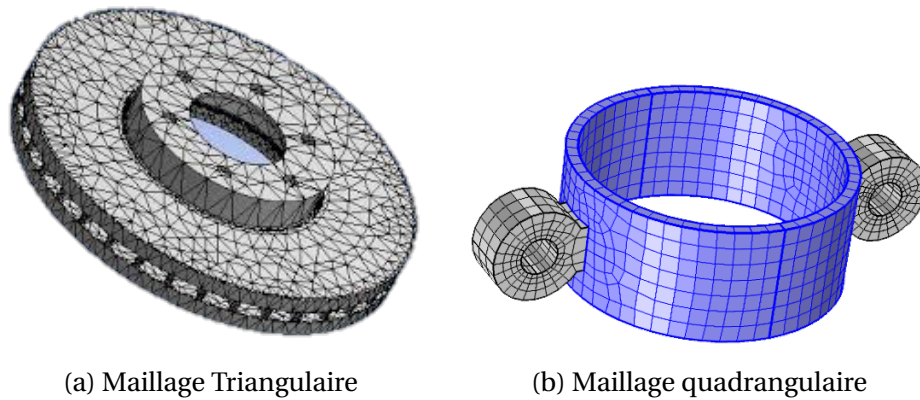


FIGURE 5.27 – Exemples de maillage.

5.5.1.3 Organisation des modèles 3D

Les contraintes architecturales nous ont permis de créer les modèles 3D préliminaires du Scalable-SPIDAR et de la station ACRS qui constitueront la base de notre travail.

Le modèle 3D de Scalable-SPIDAR se compose de deux grands éléments créés par la technique de représentation par arbre de construction CSG :

- Le corps qui est composé de plusieurs boîtes, des fils sous forme de courbes Splines et des 8 anneaux pour l'accrochage des 8 fils à la structure cubique du corps du Scalable-SPIDAR.

La figure 5.28 montre le modèle 3D du Scalable-SPIDAR :

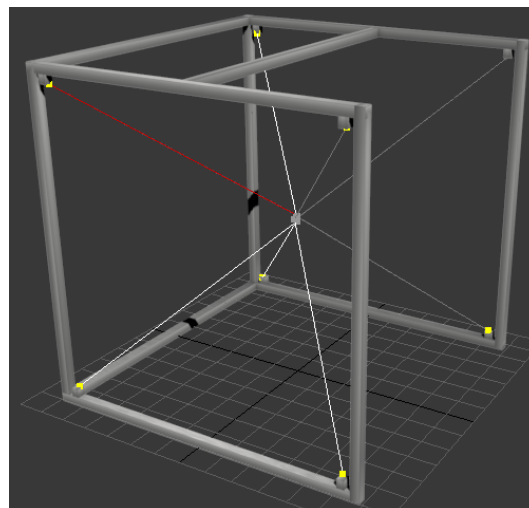


FIGURE 5.28 – Modèle 3D du Scalable-SPIDAR

- Une balle qui représente l'effecteur du Scalable-SPIDAR qui est à son tour composée par une sphère et 4 anneaux pour l'accrochage des fils à la balle selon la configuration décrite par la figure 5.28.

La figure 5.29 décrit le modèle 3D de l'effecteur du Scalable-SPIDAR.

Quant au modèle 3D de la station ACRS, il a été créé par les techniques de représentation par arbre de construction CSG et de maillage. Celui-ci se compose de :

- La base de la station qui est composée par un cube et une boîte éditée par la technique de maillage pour représenter le pupitre de commande. De plus, la base possède égale-

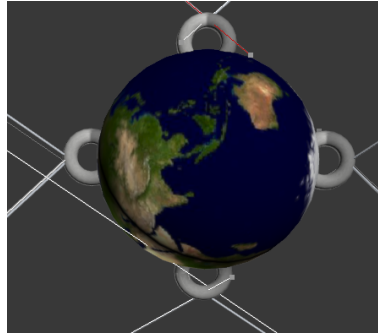


FIGURE 5.29 – Modèle 3D de l'effecteur du Scalable-SPIDAR

ment 4 roues dont chacune est créée à l'aide d'un anneau, deux cylindres et trois boîtes dont deux sont éditées par la technique de maillage.

La figure 5.30 décrit les modèles 3D de la base de la station robotisée ACRS ainsi que de sa roue.



FIGURE 5.30 – Modèles 3D de la base de la station robotisée et sa roue

- ❑ Le magasin de la Station ACRS est composé de six boîtes pour délimiter le volume du magasin et de cinq boîtes éditées par maillage pour représenter les étagères dont les emplacements de chacune sont modélisés par une boîte trouée par sept cylindres. La figure 5.31 illustre la vue d'ensemble de magasin du robot ainsi que l'étagère de pièces modélisées :
- ❑ L'articulateur de la station qui est constituée de trois boîtes pour représenter les trois axes de translation, trois joints dont chacun est modélisé aussi par un ensemble de boîtes et une pince. Cette dernière est la composition de cinq boîtes dont deux sont éditées par l'opération de soustraction d'un cylindre. La figure 5.32 montre la vue d'ensemble de l'articulateur de la station ACRS et la pince de pièce modélisés :
- ❑ Le chariot de pièces qui est composé par deux boîtes superposées l'une sur l'autre par le biais de 4 cylindres. La boîte hôte de ce chariot est trouée partiellement par 4 cylindres pour définir l'emplacement de ces 4 pièces. Le chariot transportant les pièces est représenté par la figure 5.33. Enfin, la figure 5.34 montre le modèle 3D complet de notre station robotisée ACRS.

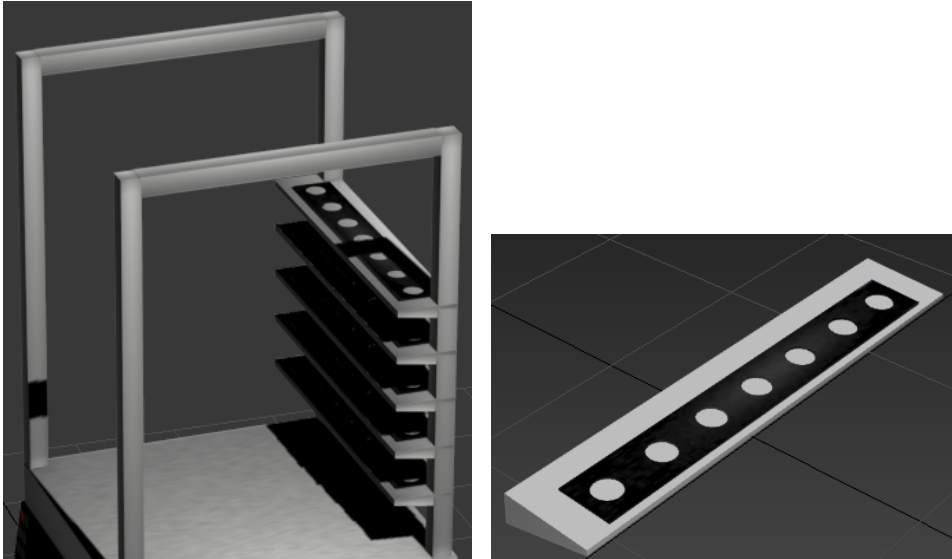


FIGURE 5.31 – Modèle 3D du magasin de la station ACRS

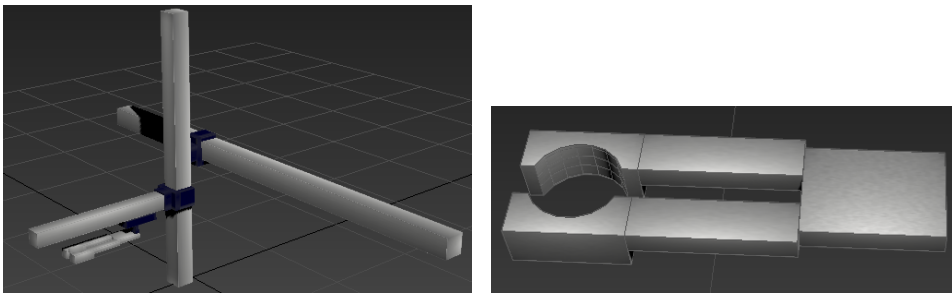


FIGURE 5.32 – Modélisation 3D de l'articulateur du robot

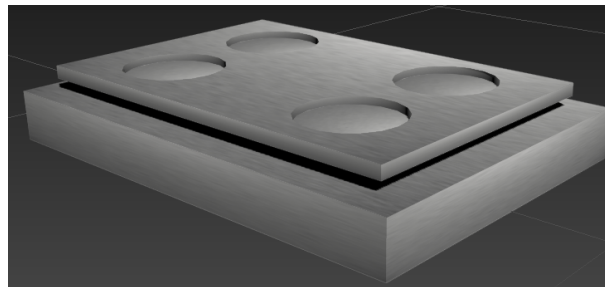


FIGURE 5.33 – Modèle 3D du chariot des pièces

5.5.2 L'Animation 3D

La deuxième étape consiste essentiellement à décrire les mouvements des différents objets créés dans la scène. Tout objet géométrique est décrit par une trajectoire spatio-temporelle.

5.5.2.1 Les contraintes fonctionnelles

Le déplacement de l'effecteur virtuel du Scalable-SPIDAR doit entraîner le déplacement des différents axes de l'articulateur de la station ACRS.

* Le déplacement de l'effecteur virtuel selon la direction X provoque le déplacement des

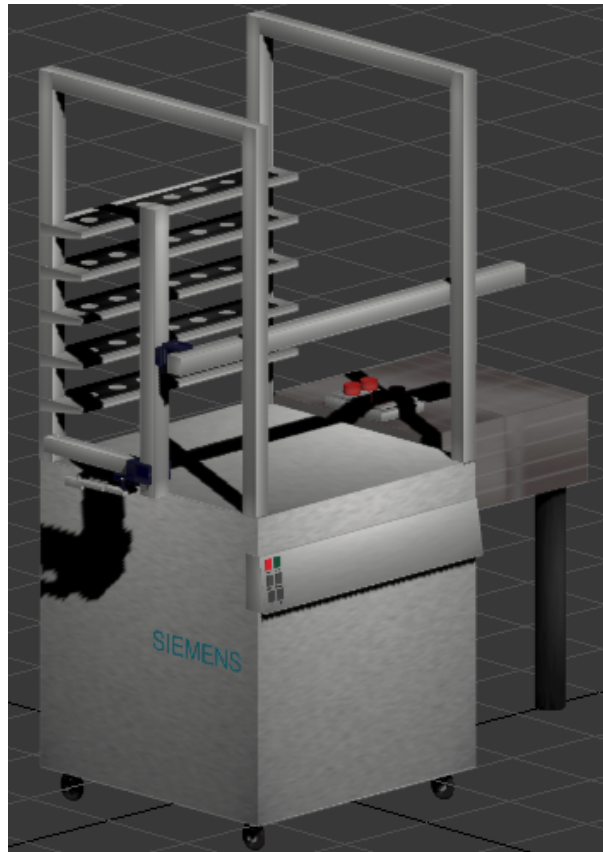


FIGURE 5.34 – Modèle 3D complet de la station ACRS

axes Y et Z de la station suivant la même direction c'est à dire X.

- * Le déplacement de l'effecteur virtuel selon la direction Z entraîne seulement le déplacement de l'axe Y de la station suivant la direction Z.
- * Le déplacement de l'effecteur selon la direction Y provoque le déplacement de la pince.

5.5.2.2 *Techniques d'animation 3D*

L'animation des modèles 3D créée est assurée par la technique des marionnettes virtuelles. Celles-ci sont manipulées de la même façon qu'une marionnette traditionnelle grâce à une structure ou « armature » articulée interne, invisible à l'œil nu. L'animation de marionnettes virtuelles requiert la mise en place d'une structure hiérarchique. L'assemblage des marionnettes virtuelles sous forme d'une structure hiérarchique permet de lier les différents objets 3D qui la composent et d'établir la logique de transmission du mouvement. Les liens établis entre les objets 3D sont qualifiés d'hiérarchiques dans la mesure où chaque composant 3D situé au plus haut niveau de la structure, contrôle tous ceux qui sont sous leur influence et non l'inverse. Le lien hiérarchique permet de concevoir les relations de pouvoir ou de dépendance entre chaque élément 3D de la structure et détermine le comportement des objets lorsqu'ils sont animés. Cette technique qui permet de décrire la dépendance d'éléments vis-à-vis d'autres éléments, convient bien à l'animation de structures articulées.

5.5.2.3 Squelettes des modèles 3D

Les squelettes de deux modèles 3D sont représentés par un ensemble d'os selon une structure descendante. En effet, le fait de faire bouger l'os père fait bouger les os fils. L'inverse n'est pas possible c'est-à-dire l'os fils ne peut pas faire bouger l'os père. En effet, le squelette de Scalable-SPIDAR (figure 5.35) a été affecté à la balle et les fils sont les éléments mobiles de cet équipement.

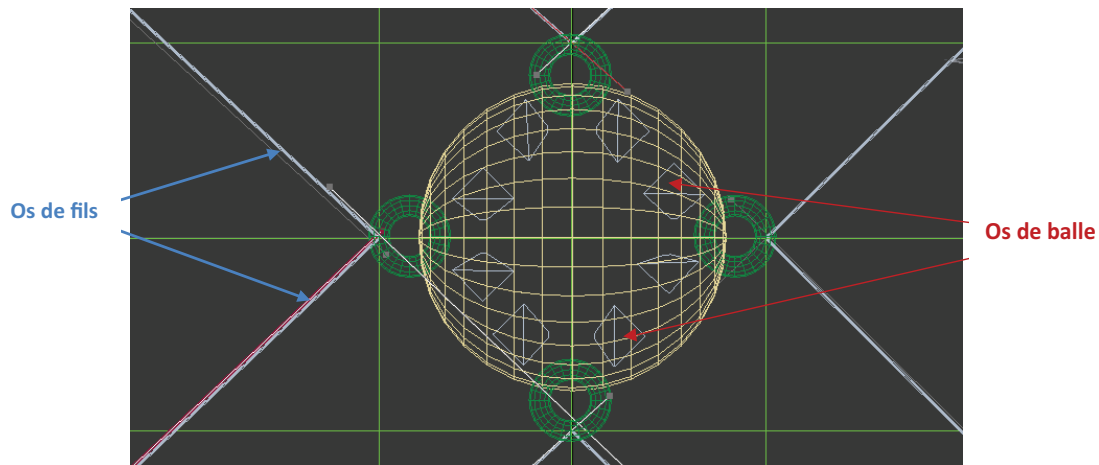


FIGURE 5.35 – Squelette de l'effecteur du Scalable-SPIDAR

Le squelette de la Station ACRS (figure 5.36) a été affecté seulement à la partie mobile celle de l'articulateur.

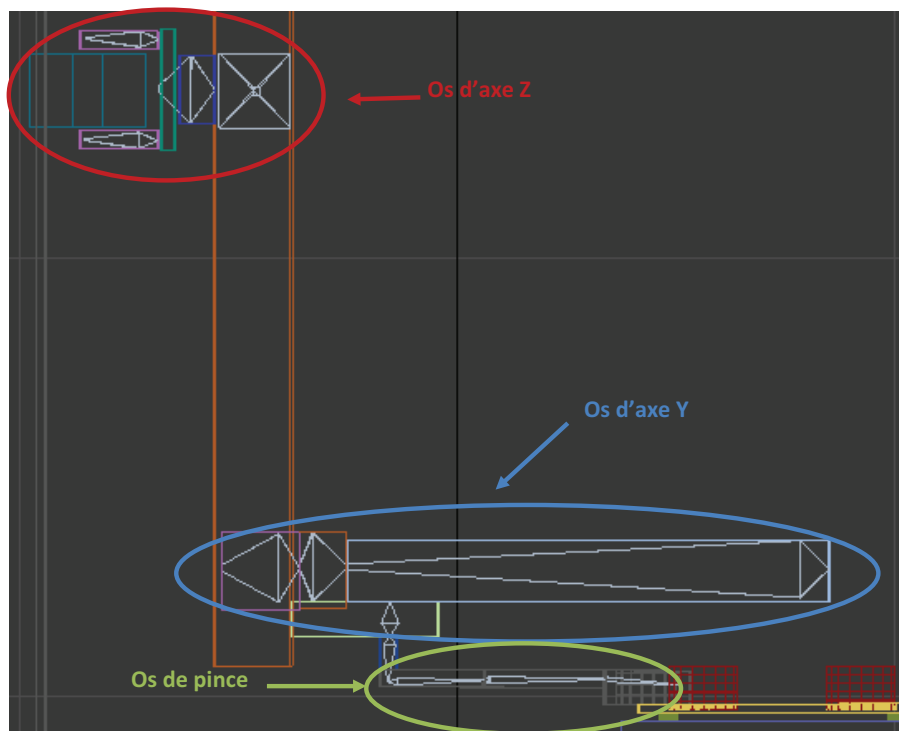


FIGURE 5.36 – Squelette de la station ACRS

5.5.3 Prototypes d'applications de réalité virtuelle

Il est temps maintenant d'évaluer la pertinence de l'approche de calibration basée sur les méthodes de traitement de données. Un premier prototype d'un simulateur virtuel qui permet de reproduire les mouvements de l'effecteur avant et après calibration a été mis en place. Il permet à l'utilisateur de distinguer la zone dans laquelle il est en train d'opérer. Le deuxième prototype est utilisé pour visualiser l'écart entre les différentes positions : référence, avant calibration et après calibration sur le modèle virtuel de la station robotisée ACRS. Cette étape est importante avant la mise en œuvre de la quasi-totalité du protocole de télétravail.

Comme le montrent les figures 5.37 et 5.38, les méthodes de calibration apportent une nette amélioration au positionnement de la pièce dans le magasin. La position corrigée est presque identique à la position de référence ce qui démontre bien l'efficacité de nos méthodes de calibration. Le prototype brièvement décrit permet de construire une représentation virtuelle des objets présents dans le site réel (Station robotisée ACRS) et de leur attribuer des repères utiles pour toutes transformations dans l'espace. Bien que cette première phase nous a servi de maquettage pour pouvoir visualiser les écarts entre les différentes positions (position de référence, position mesurée et position corrigée), il nous reste encore à établir les liens entre ce prototype virtuel et la station ACRS par le biais d'une partie d'acquisition de données. Cette étape préliminaire est nécessaire pour juger la qualité de notre approche et vérifier la faisabilité de la procédure de calibration.

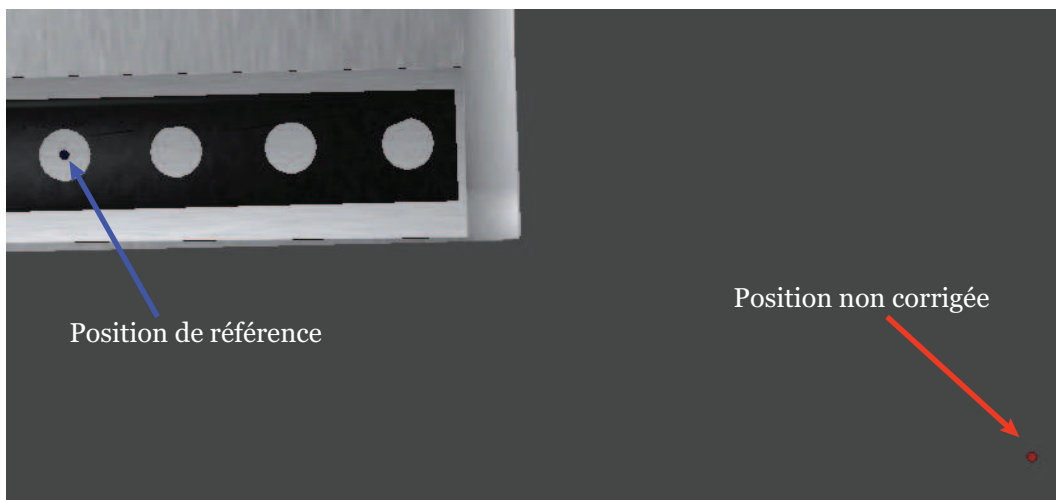


FIGURE 5.37 – Ecart entre la position de référence et la position mesurée par Scalable-SPIDAR avant calibration

5.6 LES PISTES DE NOTRE ACTION

Afin d'améliorer encore davantage les performances de la procédure de calibration et obtenir la possibilité des traitements plus rapides lors d'une utilisation en temps réel nous nous proposons d'implémenter les méthodes de calibration dans un circuit de type FPGA. Le développement de la solution technique est abordé selon une démarche descendante (Top-Down). Cette démarche permet de translater les spécifications systèmes vers les spécifications circuits. Le développement de la solution doit s'effectuer dans un

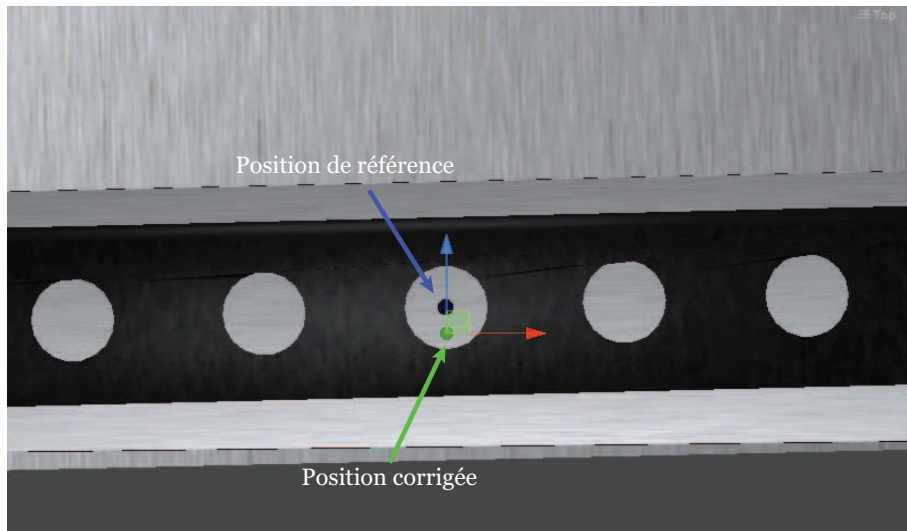


FIGURE 5.38 – Ecart entre la position de référence et la position obtenue après calibration

environnement unifié supportant l'ensemble de la hiérarchie de conception. Le langage VHDL est le candidat que nous avons retenu pour cela. Il permettra la description complète du système correspondant à la méthode de calibration avec ses blocs électroniques (modèles comportementales ou structurels).

Matlab-Simulink plus XSG (Xilinx System Generator) offre un environnement idéal de prototypage rapide des algorithmes de traitement de données. Il s'agit d'une boîte à outils développée par Xilinx pour être intégrée dans l'environnement MATLAB-Simulink qui permet à l'utilisateur de réaliser des systèmes hautement parallèles pour FPGA. Les modèles créés sont affichés sous forme de blocs et peuvent être connectés aux autres blocs et autres boîtes à outils de MATLAB. Une fois le système terminé, le programme VHDL généré par le système générateur de Xilinx XSG retrace exactement le comportement observé sous MATLAB. Le choix de cet outil dans notre cas semble être une bonne solution. En effet, étant donné que le développement des méthodes de calibration s'est effectué sous MATLAB, nous avons jugé intéressant d'intégrer cette boîte à outils pour compléter la chaîne menant à une implémentation matérielle (figure 5.39).

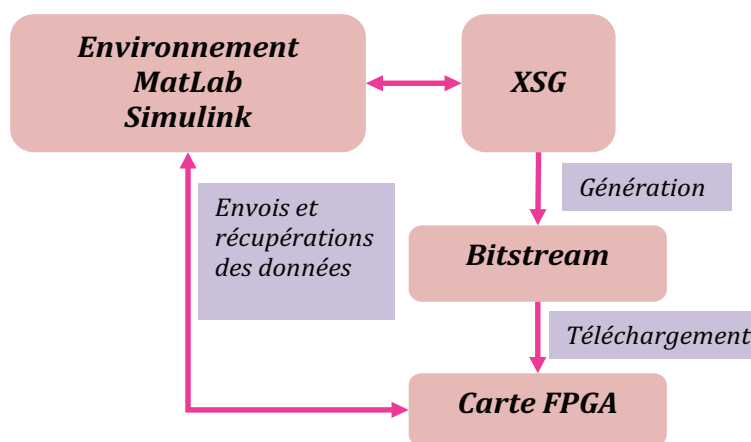


FIGURE 5.39 – Méthode de développement à l'aide du Système générateur de Xilinx (XGS)

Le système de calibration devant être simulé et vérifié plusieurs fois et rapidement pendant toute la phase de développement. En effet, il est plus facile d'analyser les résul-

tats avec MATLAB qu'avec les outils souvent associés au VHDL, comme Modelsim. Quand le prototype fonctionne convenablement, la migration vers la plateforme matérielle pour des tests sur le terrain est rapide.

CONCLUSION

Dans ce chapitre, nous avons évalué deux différentes techniques pour calibrer au mieux le Scalable-SPIDAR par rapport à une référence donnée par un système de tracking optique ARTtrack1 plus précis. Nous avons décrit une procédure d'initialisation multimodale pour accomplir l'initialisation du Scalable-SPIDAR d'une manière très précise. Nous avons ensuite, procédé à la calibration du Scalable-SPIDAR, à l'aide de deux techniques de régression non linéaire : les réseaux de neurones et les SVR. Les résultats de calibration par SVR permettent de garantir la bonne convergence du réseau de neurones vers un minimum global.

Cette méthode de calibration a été ensuite évaluée par la mise en place de deux prototypes d'applications de réalité virtuelle. Ces applications nous ont permis de mettre en exergue l'apport des méthodes de traitement de données sur la précision absolue du Scalable-SPIDAR.

Une dernière piste d'action a été explorée afin d'améliorer encore davantage les performances de la procédure de calibration et d'obtenir la possibilité des traitements plus rapides lors d'une utilisation en temps réel. L'intégration de l'outil système générateur Xilinx permettra le prototypage rapide de la solution matérielle.

Conclusion Générale et Perspectives

Nos travaux de recherche ont porté sur la proposition d'une procédure de calibration qui permet de garantir la précision et la continuité des données retournées par les interfaces d'interaction en réalité virtuelle. Notre recherche s'est axée autour de deux domaines de compétences : la réalité virtuelle et le traitement de données.

Le sujet s'articule autour du triptyque suivant : réalité virtuelle, protocoles et techniques de calibration couvrant la totalité d'une procédure de calibration en partant des interfaces même, jusqu'à l'évaluation des performances de la procédure même par l'utilisateur via une application. La qualité et la continuité de l'interaction en réalité virtuelle étant tributaire du système de tracking utilisé, il est indispensable de s'intéresser de près au traitement de données en provenance du système tracking.

L'analyse bibliographique a été axée sur trois pistes. Nous avons abordés dans un premier temps, le domaine de la réalité virtuelle, incluant les définitions utilisées, ainsi que les différents domaines d'application. Ensuite, nous avons étudié les différentes interfaces matérielles utilisées en réalité virtuelle pour faire de l'interaction. Nous avons mis en lumière, leurs avantages et inconvénients ainsi que les contraintes qu'ils imposaient. Enfin, nous avons présenté les protocoles expérimentaux et techniques de calibration de ces interfaces.

A l'issue de cette analyse nous avons pu remarquer quelques points. Nous avons remarqué qu'il est crucial de bien caractériser les systèmes de tracking à l'aide d'un protocole rigoureux avant de choisir la technique de calibration. A la revue des différentes recherches, nous nous sommes rendu compte qu'il n'existait pas de technique générale ou efficace plus que les autres, mais que ne devons faire le choix de technique cas par cas, à la lumière des résultats de caractérisation.

Les contributions de la thèse peuvent être classées en deux sous parties traitant chacune un aspect précis d'un système de réalité virtuelle.

La première porte sur le développement d'un protocole rigoureux de caractérisation d'un système de tracking peu précis par rapport à un autre système plus précis. Ce protocole a été conçu pour mettre en exergue les problèmes de conception mécanique inhérents au Scalable-SPIDAR. Ce protocole est original dans la mesure où il repose sur les techniques de réalité virtuelle pour récolter les données de position sur un volume de travail d'un mètre cube de façon quasi-statique et bien répartie sur tout l'espace de travail.

Une deuxième contribution consiste à l'utilisation des techniques de régression non linéaires pour corriger les erreurs de position. Suite à la collecte de différentes mesures, nous avons recouru aux techniques de régression et plus particulièrement aux réseaux de neurones et à la régression à vecteurs de support SVR qui se présentent comme un moyen puissant de calibration. Nous avons expérimentées deux méthodes, la première étant la mise en œuvre d'un réseau de type "perceptron multicouche" tandis que la deuxième était l'utilisation des machines à vecteurs de support pour l'estimation de la fonction de

régression. Dans le deux méthodes nous avons abouti à une nette diminution de l'erreur absolue de position sur la totalité de l'espace de travail en passant d'une plage d'erreur de l'ordre de 273mm à environ 15mm . Il apparaît à la vue de l'expérimentation que la calibration à l'aide des SVRs est une très bonne solution pour la calibration des systèmes n'ayant pas un modèle de connaissance à priori. C'est une voie de recherche à prendre en considération pour un travail futur concentrant la calibration des interfaces utilisées en réalité virtuelle.

Cette contribution a été couronnée par la mise en œuvre de deux prototypes d'application de réalité virtuelle permettant d'évaluer la pertinence de techniques utilisées. Un premier prototype d'un simulateur virtuel du Scalable-SPIDAR a été réalisé pour pouvoir retracer les trajectoires empruntés par le Scalable-SPIDAR et les comparer à des trajectoires de référence.

Le deuxième prototype s'inscrit dans un cadre plus large : le télétravail. Bien que la première phase nous a servi de maquettage pour pouvoir visualiser les écarts entre les différentes positions, il nous reste à développer les liens entre le prototype virtuel et la station robotisée ACRS par une partie d'acquisition des données. Cette étape préliminaire est nécessaire pour juger de la qualité de notre approche et vérifier la faisabilité de la procédure de calibration.

L'ensemble de ce document témoigne de la volonté de mettre au point une méthode de calibration souple et efficace qui permet de garantir la qualité et la précision des données issues des interfaces d'interaction en réalité virtuelle. En résumé, le travail mené présente une démarche originale et rarement utilisée pour la calibration des interfaces de réalité virtuelle. L'apport des méthodes de traitement de données est pertinent, tant au niveau matériel que logiciel, et permet de garantir à la fois la précision de données et par conséquent la qualité de la continuité de l'interaction en réalité virtuelle.

Cette thèse a été le fruit d'une collaboration entre deux domaines de recherche : le traitement de données et la réalité virtuelle. 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 on est la preuve. La procédure de calibration proposée est flexible et pourrait être transposée à d'autres systèmes de tracking par exemple les autres versions des interfaces à câbles telles que le SPIDAR I, II, SPIDAR-G et SPIDAR-8. Le protocole expérimental a été seulement utilisé pour évaluer les erreurs statiques, mais il est intéressant d'ajouter plus d'options au protocole pour pouvoir évaluer les erreurs dynamiques liées à la présence de l'utilisateur dans la boucle d'interaction, afin d'améliorer encore les performances de notre système de tracking. De nombreuses améliorations peuvent être apportées au Scalable-SPIDAR du point de vue de sa calibration. En effet, en utilisant une approche de tracking hybride c'est à dire le recours au système du tracking optique, il est possible d'effectuer une re-calibration du Scalable-SPIDAR en cours d'utilisation. De plus, les méthodes de calibration développées peuvent notamment être robustes, car à l'heure actuelle, les performances observées sont issues des données prises lors d'un fonctionnement ordinaire. Il serait intéressant de pouvoir tester ces méthodes dans des situations extrêmes (panne subite de l'un des moteurs du Scalable-SPIDAR, coupure d'un câble...). Une autre piste d'action d'application prometteuse est l'implémentation matérielle des méthodes de calibration dans un circuit de type FPGA. Ceci-permettra des traitements beaucoup plus rapides lors d'une utilisation temps réel.

Bibliographie

- 2003, *ARTrack1 & DTrack*, Advanced Realtime Tracking GmbH, version 1.18 éd.. 103
- ALLEN, B. D., G. BISHOP et G. WELCH. 2001, «Tracking : Beyond 15 Minutes of Thought», *Siggraph 2001*, p. 117. 53
- ARNALDI, B., P. FUCHS et J. TISSEAU. 2003, *Le traité de la réalité virtuelle*, Le Presse de l'école des Mines de Paris. 8
- ART ADVANCED REALTIME TRACKING, G. 2016, «ARTtrack1», URL <http://www.ar-tracking.com/>. v, 20, 21
- BAJURA, M., H. FUCHS et R. OHBUCHI. 1992, «Merging Virtual Objects with the Real World : Seeing Ultrasound Imagery Within the Patient», dans *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '92, ACM, New York, NY, USA, ISBN 0-89791-479-1, p. 203–210. URL <http://doi.acm.org/10.1145/133994.134061>. 11
- BEBIS, G. et M. GEORGIOPOULOS. 1994, «Feed-forward neural networks», . 77
- BERGAMASCO, M. et C. A. AVIZZANO. 1997, «Virtual environment technologies in rehabilitation», dans *Proceedings of IEEE-RSJ-SICE Symposium on Robot and Human Interactive Communication*. 29
- BIRGLEN, L., C. GOSSELIN, N. POULIOT, B. MONSARRAT et T. LALIBERTE. 2002, «SHaDe, a new 3-DOF haptic device», . vi, 31
- BIRKFELLNER, W., F. WATZINGER, F. WANSCHITZ, R. EWERS et H. BERGMANN. 1998, «Calibration of tracking systems in a surgical environment», . vi, 46, 47, 48
- BISHOP, C. M. 1995, *Neural Networks for Pattern Recognition*, Oxford University Press, Inc., New York, NY, USA, ISBN 0198538642. 79, 89
- BOUDOIN, P. 2010, *L'interaction 3D adaptative : une approche basée sur les méthodes de traitement de données multi-capteurs*, thèse de doctorat, Université d'Evry-Val d'Essonne. 111, 122
- BOUDOIN, P., H. MAAREF, S. OTMANE et M. MALLEM. 2010, «{SPIDAR} Calibration based on Neural Networks versus Optical Tracking», dans *Artificial Neural Networks and Intelligent Information Processing, Proceedings of the 6th International Workshop on Artificial Neural Networks and Intelligent Information Processing, {ANNIP} 2010, In conjunction with {ICINCO} 2010, Funchal, Madeira, Portugal, June 2010*, édité par K. Madani, {INSTICC} Press, ISBN 978-989-8425-03-4, p. 87–98. 105

- BRIGGS, W. 1999, «Magnetic calibration by tetrahedral interpolation», dans *Proceedings of NIST-ASME Industrial Virtual Reality Symposium*, vol. MED-Vol. 9, Chicago, p. 27–32. [41, 57](#)
- BROYDEN, C. G. 1970, «The Convergence of a Class of Double-rank Minimization Algorithms : 2. The New Algorithm », *IMA Journal of Applied Mathematics*, vol. 6, n° 3, p. 222–231. URL <http://imamat.oxfordjournals.org/content/6/3/222.abstract>. [81](#)
- BRYSON, S. 1992, «Measurement and calibration of static distortion of position data from 3d trackers», dans *Proc. SPIE Conf. Stereoscopic Displays and Applications III*, p. 244–255. [40, 57, 58](#)
- BUOGUILA, L., M. ISHII et M. SATO. 2000, «Multi-modal Haptic Device for Large-scale Virtual Environments», dans *Proceedings of the Eighth ACM International Conference on Multimedia*, MULTIMEDIA '00, ACM, New York, NY, USA, ISBN 1-58113-198-4, p. 277–283. URL <http://doi.acm.org/10.1145/354384.354506>. [32](#)
- BURDEA, G. et P. COIFFET. 1993, *La réalité virtuelle*, Hermès. [7](#)
- BURDEA, G. C. et P. COIFFET. 2003, *Virtual reality technology*, 2^e éd., John Wiley & Sons, Inc, New York, NY, USA, 444 p.. [v, 16, 17, 18](#)
- CENTRE DE LA RÉALITÉ VIRTUELLE DE LA MÉDITERRANÉE CRVM. 2016, «Sas-cube de marseille.», . [26](#)
- CHESTER, D. 1990, «Why two hidden layers are better than one», dans *Proceeding of IEEE International Joint Conference on Neural Networks*, Washington, p. 265–268. [78](#)
- CHUANG, C.-C., S.-F. SU, J.-T. JENG et C.-C. HSIAO. 2002, «Robust support vector regression networks for function approximation with outliers», . [89](#)
- CLARTE. 2016, «SAS cube», URL <http://www.clarte.asso.fr/realite-virtuelle.php/les-equipements-du-centre-de-realite-virtuelle.html>. [vi, 27](#)
- COIFFET, P. 1996, «Réalité virtuelle : application à la robotique», *Techniques de l'Ingénieur*, vol. 12, n° R7734, p. 1–8. [v, 14](#)
- CRAIG, A. B., W. R. SHERMAN et J. D. WILL. 2009, *Developing Virtual Reality Applications : Foundations of Effective Design*, Morgan Kaufmann, ISBN 0080959083, 448 p.. [v, 10, 11, 12, 13, 14](#)
- CRUZ-NEIRA, C., D. J. SANDIN, T. A. DEFANTI, R. V. KENYON et J. C. HART. 1992, «The CAVE : Audio Visual Experience Automatic Virtual Environment», *Commun. ACM*, vol. 35, n° 6, p. 64–72. URL <http://doi.acm.org/10.1145/129888.129892>. [26](#)
- CYBENKO, G. 1989, «Approximation by superpositions of a sigmoidal function», *Mathematics of Control, Signals and Systems*, vol. 2, n° 4, p. 303–314. URL <http://dx.doi.org/10.1007/BF02551274>. [78](#)
- CYBER GLOVES SYSTEMS. 2016, «CyberGlove», URL <http://www.cyberglovesystems.com/cyberglove-ii>. [v, 22, 23](#)

- CZERNUSZENKO, M. 1998, «Line of sight method for tracker calibration in projection-based VR systems», *Proc. 2nd International Immersive Projection Technology Workshop*. 56
- DAY, J. S., G. A. DUMAS et D. J. MURDOCH. 1998, «Evaluation of a long-range transmitter for use with a magnetic tracking device in motion analysis», *Journal of Biomechanics*, vol. 31, n° 10, p. 957–961. vi, 42, 43
- DAY, J. S., D. J. MURDOCH et G. A. DUMAS. 2000, «Calibration of position and angular data from a magnetic tracking device», *Journal of Biomechanics*, vol. 33, n° 8, p. 1039–1045, ISSN 00219290. 42
- DREYFUS, G. 1998, «Les réseaux de neurones», vol. 51, n° septembre. 73
- DREYFUS, G., J.-M. MARTTINEZ, M. SAMUELIDES, M. GORDON, F. BADRAN, S. THIRIA et L. HÉRAULT. 2008, «Réseaux de Neurones -Ch. 1», *Apprentissage statistique*, p. 449. vii, 71, 81
- DRUCKER, H., C. J. C. BURGESS, L. KAUFMAN, A. SMOLA et V. VAPNIK. 1997, «Support vector regression machines», *Advances in Neural Information Processing Systems*, vol. 1, p. 155–161. 82
- ELLIS, S. R. 1991, «Nature and origins of virtual environments : a bibliographical essay», . 7
- ELMAN, J. L. 1990, «Finding structure in time», *Cognitive Science*, vol. 14, n° 2, p. 179–211. 68
- EMMANUEL, B. 2005, *Contribution à l'étude d'interfaces portables à retour d'efforts pour la réalité virtuelle*, Thèse de doctorat, Université de Poitiers. 30
- FAKESPACE. 1994, «Binocular Omni-Orientation Monitor BOOM», URL <http://www-cdr.stanford.edu/html/DesignSpace/sponsors/boom.html>. v, 19
- FAULRING, E., J. COLGATE et M. PESHKIN. 2004, «A high performance 6-DOF haptic Co-bot», *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 2, n° April, p. 1–6. vi, 31, 32
- FIFTH DIMENSION TECHNOLOGIES. 2016, «5dt data glove ultra serie», URL <http://www.5dt.com>. v, 22
- FISCHER, G. S. 2005, *Electromagnetic Tracker Characterization and Optimal tool design*, thèse de doctorat, Johns Hopkins University. v, 18, 19, 60
- FRANTZ, D. D., A. D. WILES, S. E. LEIS et S. R. KIRSCH. 2003, «Accuracy assessment protocols for electromagnetic tracking systems», *Physics in medicine and biology*, vol. 48, n° 14, p. 2241. URL <http://iopscience.iop.org/0031-9155/48/14/314>. vi, 49, 50
- FRIEDRICHS, F. et C. IGEL. 2005, «Evolutionary tuning of multiple SVM parameters», *Neurocomputing*, vol. 64, p. 107–117. URL <http://www.sciencedirect.com/science/article/pii/S0925231204005223>. 89
- FUCHS, P. et G. MOREAU. 2003, *Le traité de la réalité virtuelle.*, seconde éd., Presse de l'Ecole des Mines. 7, 8

- FUCHS, P., G. MOREAU et P. GUITTON. 2011, *Virtual Reality : Concepts and Technologies*, 1^{re} éd., CRC Press, Inc., Boca Raton, FL, USA, ISBN 9780415684712. vi, 29
- FUNAHASHI, K. 1989, «On the Approximate Realization of Continuous Mappings by Neural Networks», *Neural Netw.*, vol. 2, n° 3, p. 183–192, ISSN 0893-6080. URL [http://dx.doi.org/10.1016/0893-6080\(89\)90003-8](http://dx.doi.org/10.1016/0893-6080(89)90003-8). 78
- GEIST, M. 2013, «Précis introductif à l'apprentissage statistique», . 83
- GEOMAGIC® HAPTIC DEVICES. 2016, «Haptic devices that add the sense of Touch to your digital world», URL <http://www.geomagic.com/en/products-landing-pages/haptic>. vi, 30
- GHADDAR, M., L. TALBI et T. A. DENIDNI. 2004, «Human body modelling for prediction of effect of people on indoor propagation channel», . 68
- GHAZISAEDY, M., D. ADAMCZYK, D. SANDIN, R. KENYON et T. DEFANTI. 1995, «Ultrasonic calibration of a magnetic tracker in a virtual reality space», URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=512494>. vi, 44, 45, 56, 105
- GOSSELIN, F. 2005, «Optimisation des interfaces haptiques : problèmes, méthodes, applications», *17ème Congrès Français de Mécanique*, p. 1–12. vi, 28, 29
- HAPTION. 2016, «La famille Virtuouse», URL <http://www.haption.com/site/index.php/fr/products-menu-fr/hardware-menu-fr/>. vi, 31
- HASEGAWA, S., I. TOSHIKI, N. HASHIMOTO, M. SALVATI, H. MITAKE, Y. KOIKE et M. SATO. 2006, «Human-scale Haptic Interaction with a Reactive Virtual Human in a Real-time Physics Simulator», *Comput. Entertain.*, vol. 4, n° 3. URL <http://doi.acm.org/10.1145/1146816.1146831>. vi, 33
- HAYASHI, Y., M. SAKATA et S. I. GALLANT. 1990, «Multi-Layer Versus Single-Layer Neural Networks and an Application to Reading Hand-Stamped Characters BT - International Neural Network Conference : July 9–13, 1990 Palais Des Congres — Paris — France», Springer Netherlands, Dordrecht, ISBN 978-94-009-0643-3, p. 781–784. 78
- HAYWARD, V., O. R. ASTLEY, M. CRUZ-HERNANDEZ, D. GRANT et G. ROBLES-DE-LA-TORRE. 2004, «Haptic interfaces and devices», *Sensor Review*, vol. 24, n° 1, p. 16–29, ISSN 0260-2288. 29
- HIMBERG, H. et Y. MOTAI. 2014, *Latency and Distortion of Electromagnetic Trackers for Augmented Reality Systems*, vol. 6, Morgan & Claypool, ISBN 9781627055079, 1–189 p.. URL <http://www.morganclaypool.com/doi/abs/10.2200/S00580ED1V01Y201404ASE012>. 43
- HIMBERG, H., Y. MOTAI et A. BRADLEY. 2012, «Interpolation Volume Calibration : A Multi-sensor Calibration Technique for Electromagnetic Trackers», *IEEE Transactions on Robotics*, vol. 28, n° 5, p. 1120–1130. vi, 43, 44
- HIRATA, Y. et M. SATO. 1992, «3-dimensional Interface Device For Virtual Work Space», . 32
- HOPFIELD, J. J. 1982, «Neural networks and physical systems with emergent collective computational abilities.», *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, n° April, p. 2554–2558. 68

- HORNIK, K. 1991, «Approximation capabilities of multilayer feedforward networks», *Neural Networks*, vol. 4, n° 2, p. 251–257. URL <http://www.sciencedirect.com/science/article/pii/089360809190009T>. 72
- HORNIK, K., M. STINCHCOMBE et H. WHITE. 1989, «Multilayer Feedforward Networks Are Universal Approximators», *Neural Netw.*, vol. 2, n° 5, p. 359–366. URL [http://dx.doi.org/10.1016/0893-6080\(89\)90020-8](http://dx.doi.org/10.1016/0893-6080(89)90020-8). 72, 78
- HORNIK, K., M. STINCHCOMBE, H. WHITE et P. AUER. 1994, «Degree of Approximation Results for Feedforward Networks Approximating Unknown Mappings and Their Derivatives», . 73
- HUANG, W. Y. et R. P. LIPPMANN. 1988, «Neural Net and Traditional Classifiers», *Neural Information Processing Systems*, p. 387–396. URL <http://papers.nips.cc/paper/26-neural-net-and-traditional-classifiers>. 78
- HUMMEL, J. B., M. R. BAX, M. L. FIGL, Y. KANG, C. MAURER, W. W. BIRKFELLNER, H. BERGMANN et R. SHAHIDI. 2005, «Design and application of an assessment protocol for electromagnetic tracking systems», *Medical Physics*, vol. 32, n° 7. vi, 42, 44
- IGATECHNOLOGIES. 2016, «Cave-Virtual room», URL http://igatec.com/en/products_services/cave. v, 26
- IKITS, M., J. BREDEPERSON, C. HANSEN et J. HOLLERBACH. 2001, «An improved calibration framework for electromagnetic tracking devices», *Proceedings IEEE Virtual Reality 2001*, p. 63–70. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=913771>. vi, 48, 58, 59
- INTERSENSE. 1999, «Technical Overview IS-900 Motion Tracking System», cahier de recherche, Intersense. v, 20
- ISHII, M., M. NAKATA et M. SATO. 1994, «Networked SPIDAR : a networked virtual environment with visual, auditory, and haptic interactions», *Presence*, vol. 3, n° January, p. 351–359. 32
- ISHII, M. et M. SATO. 1993, «A 3D interface device with force feedback : a virtual work space for pick-and-place tasks», . 32
- J HUMMEL AND M FIGL AND W BIRKFELLNER AND M R BAX AND R SHAHIDI AND C R MAURER JR AND H. 2006, «Evaluation of a new electromagnetic tracking system using a standardized assessment protocol», *Physics in Medicine and Biology*, vol. 51, n° 10, p. N205. URL <http://stacks.iop.org/0031-9155/51/i=10/a=N01>. 43
- JIMENO, A. et A. PUERTA. 2007, «State of the art of the virtual reality applied to design and manufacturing processes», *The International Journal of Advanced Manufacturing Technology*, vol. 33, n° 9, p. 866–874. URL <http://dx.doi.org/10.1007/s00170-006-0534-2>. 12
- JUHOS, I., L. MAKRA et B. TÓTH. 2009, «The behaviour of the multi-layer perceptron and the support vector regression learning methods in the prediction of NO and NO2 concentrations in Szeged, Hungary», *Neural Computing and Applications*, vol. 18, n° 2, p. 193–205. URL <http://dx.doi.org/10.1007/s00521-007-0171-1>. 89

- KASKI, S., T. HONKELA, K. LAGUS et T. KOHONEN. 1998, «WEBSOM - Self-organizing maps of document collections», *Neurocomputing*, vol. 21, n° 1-3, p. 101–117. 68
- KECMAN, V. 2001, *Learning and Soft Computing : Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, MIT Press, Cambridge, MA, USA, ISBN 0262112558. 78
- KECMAN, V. 2005a, *Studies in Fuzziness and Soft Computing , Volume 172*, vol. 18. 83
- KECMAN, V. 2005b, «Support Vector Machines – An Introduction BT - Support Vector Machines : Theory and Applications», Springer Berlin Heidelberg, ISBN 978-3-540-32384-6, p. 1–47. URL http://dx.doi.org/10.1007/10984697_{_}1. 91
- KEERTHI, S. S. et C.-J. LIN. 2003, «Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel», *Neural Comput.*, vol. 15, n° 7, p. 1667–1689. URL <http://dx.doi.org/10.1162/08997660321891855>. 90
- KEVIN, K., A. HEILBRUN et S. BARBARA. 1989, «Virtual Reality : an Interview with Jaron Lanier», *Whole Earth Review Fall*, vol. 64, p. 108–120. 7
- KIM, S., S. HASEGAWA, Y. KOIKE et M. SATO. 2002, «Tension based 7-DOF force feedback device : SPIDAR-G», . 32
- KINDRATENKO, V. 1999, «Calibration of electromagnetic tracking devices», *Virtual Reality*, vol. 4, p. 139–150, ISSN 1359-4338. 40, 58
- KINDRATENKO, V. et A. BENNETT. 2000, «Evaluation of Rotation Correction Techniques for Electromagnetic Position Tracking Systems», dans *Virtual Environments 2000 SE - 3*, édité par J. Mulder et R. van Liere, Eurographics, Springer Vienna, ISBN 978-3-211-83516-6, p. 13–22. 41, 58
- KINDRATENKO, V. V. 2000, «A survey of electromagnetic position tracker calibration techniques», *Virtual Reality*, vol. 5, n° 3, p. 169–182. x, 41, 55, 57, 63
- KINDRATENKO, V. V. et W. R. SHERMAN. 2005, «Neural network-based calibration of electromagnetic tracking systems», *Virtual Reality*, vol. 9, n° 1, p. 70–78. vi, vii, x, 41, 61, 62, 63, 127
- KUNZLER, U. et C. RUNDE. 2005, «Kinesthetic haptics integration into large-scale virtual environments», . vi, 34, 95
- KWOK, T.-Y. et D.-Y. YEUNG. 1997, «Constructive algorithms for structure learning in feedforward neural networks for regression problems», . 79
- KŮRKOVÁ, V. 1992, «Kolmogorov's theorem and multilayer neural networks», *Neural Networks*, vol. 5, n° 3, p. 501–506. URL <http://www.sciencedirect.com/science/article/pii/0893608092900128>. 78
- LÉCUYER, A. 2001, *Contribution à l'étude des retours haptique et pseudo-haptique et de leur impact sur les simulations d'opérations de montage/démontage en aéronautique*, Thèse de doctorat, Université Paris XI. 28
- LEVENBERG, K. 1944, «A method for the solution of certain non-linear problems in least squares», *Quarterly Journal of Applied Mathematics*, vol. II, n° 2, p. 164–168. 81

- LIN, K. M. et C. J. LIN. 2003, «A study of reduced support vector machines», *IEEE Transactions on Neural Networks*, vol. 14, n° 6, p. 1449–1459. [90](#)
- LIVINGSTON, M. A. et A. STATE. 1997, «Magnetic tracker calibration for improved augmented reality registration», *Presence : Teleoperators & Virtual Environments*, vol. 6, n° 5, p. 532–546. [vi](#), [45](#), [46](#), [56](#)
- MARQUARDT, D. 1963, «An Algorithm for Least-Squares Estimation of Nonlinear Parameters», *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, n° 2, p. 431–441. URL <http://dx.doi.org/10.1137/0111030>. [81](#)
- MCCULLOCH, W. S. et W. PITTS. 1943, «A logical calculus of the ideas immanent in nervous activity», *The Bulletin of Mathematical Biophysics*, vol. 5, n° 4, p. 115–133. [69](#)
- MCKENNA, B. 2000, «Life, love and the pursuit of virtual reality : An interview with inventor Jaron Lanier», *Arts & Ideas magazine*, p. Vo. 1, No. 3. [8](#)
- MICROSOFT. 2016, «Kinect for Windows Sensor Components and Specifications», URL <https://msdn.microsoft.com/en-us/library/jj131033.aspx>. [v](#), [25](#)
- MIHELJ, M., D. NOVAK et S. BEGUŠ. 2014, *Virtual Reality Technology and Applications*, vol. 68, Springer Netherlands, ISBN 978-94-007-6909-0, 1381–1394 p.. URL <http://link.springer.com/10.1007/978-94-007-6910-6>. [15](#)
- MILGRAM, P., H. TAKEMURA, A. UTSUMI et F. KISHINO. 1994, «Augmented Reality : A class of displays on the reality-virtuality continuum», *SPIE (International Society of Optics and Photonics)*, vol. 2351, n° Telemanipulator and Telepresence Technologies, p. 282–292. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.6861&rep=rep1&type=pdf>. [v](#), [9](#)
- MOREIRA, A. H. J., S. QUEIROS, N. F. RODRIGUES, A. C. M. PINHO, J. C. FONSECA et J. L. VILACA. 2014, «Electromagnetic tracker feasibility in the design of a dental superstructure for edentulous patients», *IEEE MeMeA 2014 - IEEE International Symposium on Medical Measurements and Applications, Proceedings*, p. 1–6. [vi](#), [51](#), [56](#)
- MUJBER, T. S., T. SZECSEI et M. S. J. HASHMI. 2004, «Virtual reality applications in manufacturing process simulation», *Journal of Materials Processing Technology*, vol. 155–156, p. 1834–1838. URL <http://www.sciencedirect.com/science/article/pii/S0924013604005618>. [12](#)
- NAKADA, K., M. NAKAMOTO, Y. SATO, K. KONISHI, M. HASHIZUME et S. TAMURA. 2003, «A Rapid Method for Magnetic Tracker Calibration Using a Magneto-Optic Hybrid Tracker BT - Medical Image Computing and Computer-Assisted Intervention», dans *MICCAI 2003 : 6th International Conference, Montréal, Canada, November 15-18, 2003. Proceedings*, édité par R. E. Ellis et T. M. Peters, Springer Berlin Heidelberg, ISBN 978-3-540-39903-2, p. 285–293. [52](#), [58](#)
- NINTENDO. 2010, «Wiimote controller», URL <http://www.nintendo.com/wii/what/controllers{#}remote>. [v](#), [24](#)
- NIXON, M. A., B. C. MCCALLUM, W. R. FRIGHT et N. B. PRICE. 1998, «The Effects of Metals and Interfering Fields on Electromagnetic Trackers», *Presence : Teleoperators and Virtual Environments*, vol. 7, n° 2, p. 204–218. URL <http://dx.doi.org/10.1162/105474698565587>. [58](#), [59](#)

- NVIS. 2016, «nVisor Head-Mounted Display», URL <http://www.nvisinc.com/product.html>. vi, 28
- OCHOA-MAYORGA, V. M., P. BOULANGER et M. GARCÍA. 2005, «Local quaternion weighted difference functions for orientation calibration on electromagnetic trackers», *IEEE CAMSAP 2005 - First International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, vol. 2005, p. 233–236. vi, 42, 43
- OLSON, D. et D. DELEN. 2008, *Advanced data mining techniques*, ISBN 9783540769163. 90
- PASCAL, V. 2003, *Modèles à noyaux à structure locale*, Thèse de doctorat, Université de Montréal. 72
- PÉRIÉ, D., A. J. TATE, P. L. CHENG et G. A. DUMAS. 2002, «Evaluation and calibration of an electromagnetic tracking device for biomechanical analysis of lifting tasks», *Journal of Biomechanics*, vol. 35, n° 2, p. 293–297. URL [http://dx.doi.org/10.1016/S0021-9290\(01\)00188-9](http://dx.doi.org/10.1016/S0021-9290(01)00188-9). vi, 49
- PRESS, W., S. TEUKOLSKY, W. VETTERLING, B. FLANNERY et E. ZIEGEL. 1987, *Numerical Recipes : The Art of Scientific Computing*, vol. 29, Cambridge University Press, ISBN 0521431085, 501 p.. URL <http://www.jstor.org/stable/1269484?origin=crossref>. 56, 81
- REED, R. 1993, «Pruning Algorithms-a Survey», *Trans. Neur. Netw.*, vol. 4, n° 5, p. 740–747, ISSN 1045-9227. URL <http://dx.doi.org/10.1109/72.248452>. 79
- RIVALS, I. et L. PERSONNAZ. 2003, «Neural-network construction and selection in nonlinear modeling», . 79
- SAMANTA, B., K. R. AL-BALUSHI et S. A. AL-ARAIMI. 2003, «Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection», *Engineering Applications of Artificial Intelligence*, vol. 16, n° 7–8, p. 657–665. URL <http://www.sciencedirect.com/science/article/pii/S0952197603001143>. 89
- SATO, M. 2002, «Development of string-based force display : SPIDAR», *In Proc. The Eighth International Conference on Virtual Systems and Multi Media VSMM*, p. 1034–1039. URL <https://core.ac.uk/display/24302768>. 32
- SCHÖLKOPF, B., A. J. SMOLA, R. C. WILLIAMSON et P. L. BARTLETT. 2000, «New Support Vector Algorithms», *Neural Comput.*, vol. 12, n° 5, p. 1207–1245, ISSN 0899-7667. URL <http://dx.doi.org/10.1162/089976600300015565>. 90
- SMISEK, J., M. JANCOSSEK et T. PAJDLA. 2011, «3D with Kinect», . 24
- SMOLA, A. J. et B. SCHÖLKOPF. 1998, «On a Kernel-Based Method for Pattern Recognition, Regression, Approximation, and Operator Inversion», *Algorithmica*, vol. 22, p. 211–231, ISSN 0178-4617. 82
- SMOLA, A. J. et B. SCHÖLKOPF. 2004, «A Tutorial on Support Vector Regression», *Statistics and Computing*, vol. 14, n° 3, p. 199–222. URL <http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88>. 87, 89
- STRICKLAND, D. 1996, «A Virtual Reality Application with Autistic Children», *Presence : Teleoperators and Virtual Environments*, vol. 5, n° 3, p. 319–329. URL <http://dx.doi.org/10.1162/pres.1996.5.3.319>. 11

- VAPNIK, V. N. 1995, *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., New York, NY, USA, ISBN 0-387-94559-8. 82
- VIRTUAL REALITY APPLICATIONS CENTER OF IOWA STATE UNIVERSITY VRAC. 2016, «C6», URL <http://www.vrac.iastate.edu/c6.php>. v, 26, 27
- VLADIMIR N. VAPNIK. 1997, «The nature of statistical learning theory», *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*. 82
- WALAIRACHT, S., K. YAMADA, S. HASEGAWA, Y. KOIKE et M. SATO. 2002, «4 + 4 Fingers Manipulating Virtual Objects in Mixed-reality Environment», *Presence : Teleoper. Virtual Environ.*, vol. 11, n° 2, p. 134–143. URL <http://dx.doi.org/10.1162/1054746021470586>. vi, 33
- WANG, X., C. YANG, B. QIN et W. GUI. 2005, «Parameter selection of support vector regression based on hybrid optimization algorithm and its application», *Journal of Control Theory and Applications*, vol. 4, p. 371–376. URL <http://link.springer.com/article/10.1007/s11768-005-0026-1>. 89
- WELCH, G. et E. FOXLIN. 2002, «Motion tracking : no silver bullet, but a respectable arsenal», . 15
- WERBOS, P. 1974, *Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Sciences*, thèse de doctorat, Cambridge, MA. 74
- WU, X. et R. H. TAYLOR. 2003, «A framework for calibration of electromagnetic surgical navigation system», . vi, 50, 52
- XSENS. 2016, «Human motion tracker MTx», URL <https://www.xsens.com>. v, 22
- ZACHMANN, G. 1997, «Distortion correction of magnetic fields for position tracking», *Proceedings Computer Graphics International*, p. 213–220,. 60
- ZACHMANN, G. 2000, «Virtual Reality in Assembly Simulation — Collision Detection , Simulation Algorithms , and Interaction Techniques», *Virtual Reality*, p. 231. vi, 42, 60
- ZELTZER, D. 1992, «Autonomy, Interaction, and Presence», *Presence : Teleoper. Virtual Environ.*, vol. 1, n° 1, p. 127–132, ISSN 1054-7460. URL <http://dx.doi.org/10.1162/pres.1992.1.1.127>. v, 8
- ZIMMERMANN, P. 2008, *Product Engineering : Tools and Methods Based on Virtual Reality*, chap. Virtual Re, Springer Netherlands, Dordrecht, ISBN 978-1-4020-8200-9, p. 277–296. v, 12, 13

Annexe A

Présentation de Virtools®

INTRODUCTION

Virtools® est un logiciel de développement et de déploiement d'application 3D, vendu par Dassault Systèmes. Il vise le domaine des jeux vidéos, du contenu 3D sur le Web, des applications 3D interactives pour l'industrie ainsi que les applications en RV/RA. Ce logiciel permet le prototypage d'application, via une programmation par blocs. L'avantage de cette approche est qu'il n'est pas nécessaire d'avoir de grandes connaissances en programmation. Il est possible d'exporter ses travaux réalisés avec Virtools® vers de multiples plateformes (PC, Mac, consoles de jeu) et sous multiples formes (exécutable, CD-ROM, DVD-ROM, player Web). En revanche, ce n'est en aucun cas un modéleur mais il est capable d'importer plusieurs formats de modèle 3D (3ds Max®, Maya®, XSI®, Lightwave®, Collada®).

LES BUILDING BLOCKS (BBs)

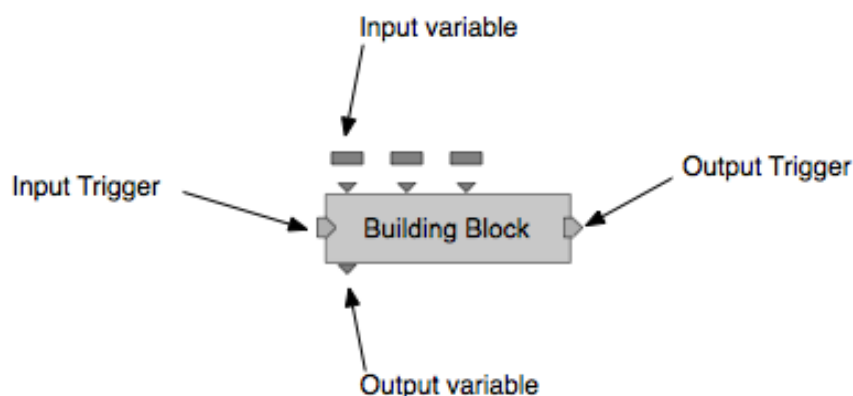


FIGURE A.1 – Représentation d'un Building Block.

Un des points forts de Virtools® se situe dans son système de développement, qui consiste à connecter des blocs appelé « Building Blocks » ou BB. Ces BBs, ont chacune un comportement spécifique qui est réalisé lorsque le BB reçoit un signal en entrée (Input trigger). Lorsque le BB a réalisé ce pour quoi il a été programmé, il émet un signal a

sa sortie (Output trigger). Certains BBs prennent en compte une ou plusieurs variables d'entrée/sortie (input/output variable). Le schéma ci-dessus (figure A.1) montre la représentation d'un BB.

Virtools® est fourni avec un certain nombre de BBs effectuant toutes les fonctions basiques (translation, rotation, ...) mais aussi quelques fonctions plus évoluées (détection de collision, calcul de point d'impact, ...). Il est cependant possible de développer soi-même des BBs, et cela de trois manières différentes.

En utilisant le Software Development Kit (SDK) Virtools® est fourni avec tous les codes sources des BBs de base, ainsi qu'avec un assistant pour créer des BBs dans l'environnement de développement Microsoft Visual Studio. En étudiant, ou même tout simplement en modifiant les sources existantes, il est très facile pour un programmeur de développer son propre BB.

En utilisant le Virtools Script Language (VSL) Au sein même de Virtools® se situe un éditeur de code VSL. Le VSL est un langage de programmation un peu plus haut niveau que le C/C++, très facile à prendre en main pour l'habitué du langage C/C++. L'éditeur, comporte toutes les avancées des derniers éditeurs de code (coloration syntaxique et écriture prédictive).

En créant une BB englobant plusieurs BBs Virtools® permet d'inclure dans un BB, un ensemble de BBs déjà connectés entre eux. L'avantage de cette technique est de clarifier très simplement le schematic. Par exemple : on passe d'un ensemble d'une trentaine de BBs, n'étant pas facilement manipulables et effectuant une seule action, à un seul BB qui lui sera très facilement manipulable.

LE SCHEMATIC ET LES SCRIPTS

Le schematic est la zone dans laquelle on insère les scripts. Ces derniers sont constitués de plusieurs BBs connectés entre eux. La figure A.2 représente un aperçu du schematic de Virtools®.

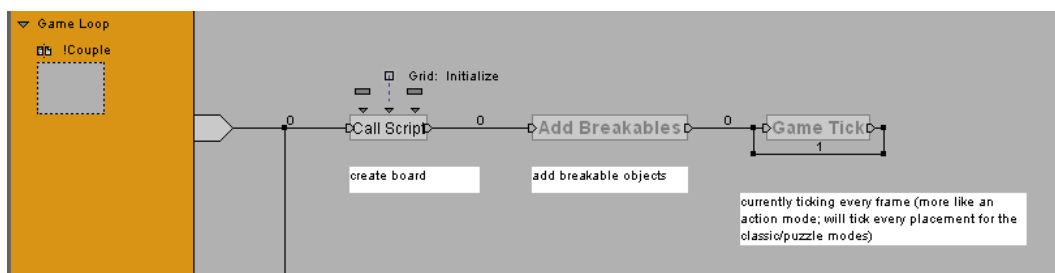


FIGURE A.2 – Le schematic de Virtools®

ATTRIBUTS SUR LES OBJETS

Dans Virtools®, des attributs peuvent être ajoutés à chaque objet présent dans la scène. Ces attributs peuvent traduire des comportements ou des propriétés pouvant définir :

- si l'objet est collisionnable

- si l'objet doit permettre de se faire projeter une ombre sur sa surface
- si l'objet doit être considéré comme un sol ou un mur
- et bien d'autres

Annexe B

Programmation sous Virtools

Une bibliothèque de modules a été créée. Elle permet le développement d'un protocole expérimental permettant la caractérisation du Scalable-SPIDAR. Ainsi par exemple, par une composition de modules nous pouvons obtenir un environnement composés d'objets à sélectionner via une technique de sélection. Nous pouvons distinguer par exemple le script SELECTION qui permet d'intégrer une technique de sélection dans l'environnement virtuel.

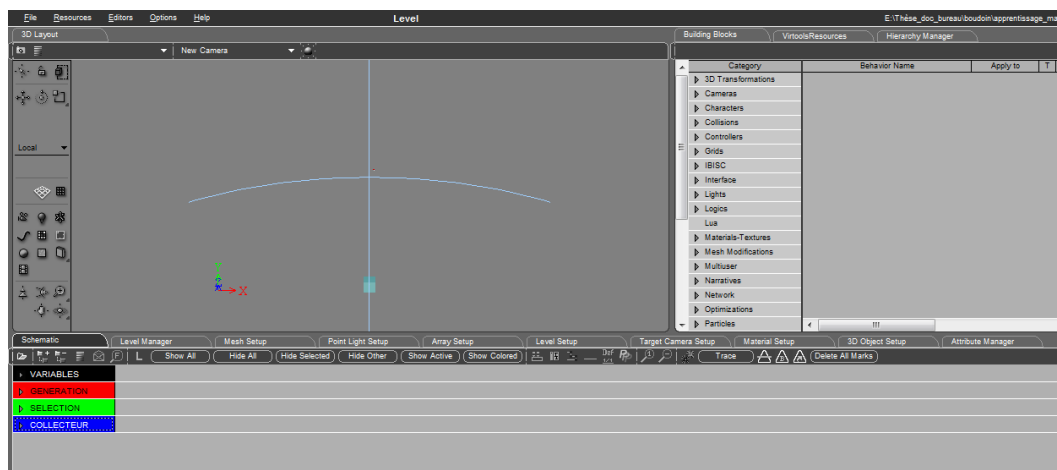


FIGURE B.1 – Les différents scripts chargés dans Virtools

