

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université SAAD DAHLAB Blida1

Faculté des Sciences

Département d'Informatique



Mémoire présenté par :

Ramdane Souhaib

Anane Imad Eddine

En vue d'obtenir le diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Informatique

Option : Ingénierie des logiciels

Thème

**Animation d'Avatars par l'analyse de mouvement :
application à l'analyse de la marche**

Soutenu le 08 Septembre 2020, devant le jury composé de :

Melle. Ykhlef Hadjer

Présidente

Mr. Hammouda Mohamed

Examineur

Mme. Abed Hafida

Promotrice

Mr. Djekoune A. Oualid

Encadreur

Mme. Zenati Henda

Encadrante

Promotion : 2019-2020

Dédicaces

A MES TRÈS CHERS PARENTS

*JE VOUS DOIS CE QUE JE SUIS AUJOURD'HUI GRÂCE À VOTRE AMOUR, À
VOTRE PATIENCE ET VOS INNOMBRABLES SACRIFICES.*

À MES CHÈRES SŒURS MAROUA ET ICHRAK

À MES ADORABLES FRÈRES: ZAKARIA, AYOUB.

À MON CHER BINÔME IMAD

À MES AMIS AVEC QUI J'AI PARTAGÉ DES MOMENTS LES PLUS AGRÉABLES.

*À TOUS CEUX QUI SONT CHERS, PROCHES DE MON CŒUR, ET À TOUS CEUX
QUI M'AIMENT ET QUI AURAIT VOULU PARTAGER MA JOIE ...*

Souhaib

Dédicaces

L'ACHÈVEMENT DE CE PROJET N'AURAIT PAS ÉTÉ POSSIBLE SANS LA PARTICIPATION ET L'AIDE DE NOMBREUSES PERSONNES DONT LES NOMS NE PEUVENT PAS TOUS ÊTRE ÉNUMÉRÉS. LEURS CONTRIBUTIONS SONT SINCÈREMENT APPRÉCIÉES ET RECONNUES. CEPENDANT, JE VOUDRAIS EXPRIMER MA PLUS PROFONDE GRATITUDE ET DÉDIER CE TRAVAIL : AUX ÊTRES LES PLUS CHERS À MON CŒUR, MA MÈRE ET MON PÈRE, QUI ONT CONSACRÉ LEUR VIE À MON ÉDUCATION ET MON BIEN ÊTRE, QUI M'AIDENT À PROGRESSER DANS LA VIE, DE PAR LEUR AMOUR ET LEUR PRÉCIEUX CONSEILS ET SOUTIENS. QUE DIEU ME LES GARDE INCHALLAH !

À MON FRÈRE BILAL ET À MES SŒURS ASMA ET MAISSA ET À MES TANTES POUR LEURS ENCOURAGEMENTS PERMANENTS, ET LEUR SOUTIEN MORAL. À MES GRANDS-MÈRES QUI NE M'ONT JAMAIS OUBLIÉ AVEC LEURS PRIÈRES ET ENCOURAGEMENTS, QU'ALLAH LEUR PROCURE BONNE SANTÉ ET LONGUE VIE. À MES MEILLEURS AMIS ZAKI, DJAMEL, OUSSAMA ET BILAL. À L'ENSEMBLE DES AMIS ET CAMARADES QUE J'AI CONNUS TOUT AU LONG DE MON CURSUS ET QUE JE REMERCIE POUR LEURS CONSEILS ET ENCOURAGEMENTS. À MON BINÔME SOUHAIB SANS LUI JE N'AURAI PAS RÉALISÉ CE TRAVAIL. À MA FAMILLE ET À TOUS CEUX QUI, DE PRÈS OU DE LOIN, ONT CONTRIBUÉ À LA RÉALISATION DE CE PROJET.

Imad

Remerciements

Avant tout, nous remercions Dieu le tout puissant de nous avoir permis de mener à bien notre mission.

Nous tenons à exprimer nos vives reconnaissances à nos encadrateurs, Dr ZENATI-HENDA Nadia et Dr DJEKOUNE A. Oualid pour leur encadrement continu et leurs remarques constructives, pour les brillantes intuitions qu'ils nous ont partagés ainsi que pour leurs précieux conseils qui ont contribué à améliorer notre travail.

Nous remercions également Pr. H. Abed pour avoir accepté d'encadrer ce travail, pour ses conseils et ses encouragements.

Nous tenons à remercier les travailleurs du Centre de Développement des Technologies Avancées (CDTA) pour l'accueil qui nous a été réservé et la mission qui nous a été confiée, particulièrement les membres de l'équipe IRVA de la division robotique pour leur aide et pour avoir mis à notre disposition le matériel nécessaire pour la réalisation de notre projet.

Nos remerciements s'adressent aussi aux membres du Jury Mlle H. Ykhlef et Mr M. Hammouda pour nous avoir honorés en consentant à juger notre travail.

Enfin, nous tenons à présenter nos remerciements les plus sincères à nos familles, nos parents, nos sœurs et nos frères pour leur soutien, encouragements et conseils tout au long de la préparation de ce mémoire.

Résumé

Dans le présent travail, nous proposons une solution qui repose principalement sur l'animation virtuelle des avatars par l'acquisition des mouvements d'un sujet humain en utilisant un réseau de capteurs. Ce dernier est basé sur une structure coopérative de plusieurs capteurs Kinect afin d'éliminer la plupart des problèmes d'occultation liés aux systèmes mono-capteur. Notre système se compose d'un module d'acquisition et de prétraitement des données squelettiques qui permet de filtrer et corriger les données récoltées de chaque capteur Kinect et d'un deuxième module qui permet de calibrer les différents capteurs utilisés deux à deux. Un troisième module a été développé pour la fusion des données squelettiques traitées pour reconstruire la posture du sujet humain à chaque frame et de reproduire ses mouvements par une animation d'un avatar dans un monde virtuel.

L'approche retenue pour notre système est basée sur l'utilisation de cinq capteurs Kinect, ce qui présente le meilleur compromis entre la précision, le réalisme de l'animation et le traitement en temps réel.

Mots clés : Capture de mouvement, Capteur Kinect, Multi-Kinect, Réalité virtuelle, Animation d'avatars, Analyse de la marche.

Abstract

In the present work, we propose a solution based mainly on the virtual animation of avatars by acquiring the movements of a human subject using a network of sensors. The latter is based on a cooperative structure of several Kinect sensors in order to eliminate most of the occultation problems associated with single-sensor systems. Our system consists of a skeletal data acquisition and pre-processing module that filters and corrects the data collected from each Kinect sensor and a second module that calibrates the different sensors used in pairs. A third module has been developed to merge the processed skeletal data to reconstruct the posture of the human subject at each frame and to reproduce his movements by animating an avatar in a virtual world.

The approach adopted for our system is based on the use of five Kinect sensors, which presents the best compromise between precision, realism of animation and real-time processing.

Keywords: Motion Capture, Kinect Sensor, Multi-Kinect, Virtual Reality, Avatar Animation, Gait Analysis.

ملخص

في هذا المشروع ، نقتراح حلاً يعتمد بشكل أساسي على العروض المتحركة الافتراضية لل Avatar من خلال التقاط الحركات البشرية باستخدام شبكة من أجهزة استشعار الحركة .

يعتمد الأخير على هيكل تعاوني للعديد من مستشعرات Kinect من أجل القضاء على معظم مشاكل الاحتجاب المرتبطة بأنظمة المستشعر الفردي. يتكون نظامنا من وحدة الحصول على البيانات الهيكلية والمعالجة المسبقة التي تقوم بتصحيح البيانات التي تم جمعها من كل مستشعر Kinect ووحدة ثانية تقوم بإيجاد العلاقات بين معالم المستشعرات المستخدمة . إلى جانب وحدة ثالثة لدمج البيانات الهيكلية التي تمت معالجتها لإعادة بناء وضعية الإنسان في كل لحظة وإعادة إنتاج حركته في عالم افتراضي ثلاثي الأبعاد من خلال تحريك Avatar.

يعتمد نظامنا على استخدام خمسة مستشعرات Kinect ، والتي تقدم أفضل حل وسط بين الدقة وواقعية العروض المتحركة والمعالجة الآتية للبيانات.

الكلمات المفتاحية: التقاط الحركة ، مستشعر Kinect ، متعدد المستشعرات ، الواقع الافتراضي ، العروض المتحركة الافتراضية ، تحليل المشي.

Table des matières

Dédicaces.....	I
Dédicaces.....	II
Remerciements	III
Résumé.....	IV
Abstract.....	V
ملخص.....	VI
Table des matières.....	VII
Liste des figures.....	XI
Liste des tableaux.....	XIII
Introduction générale	1
Partie I Systèmes de capture de mouvement.....	5
Chapitre 1 Système de capture de mouvement : Etat de l'art	6
1 Introduction.....	7
2 Analyse du mouvement humain	7
2.1 Mouvement humain	8
2.1.1 Définitions	8
2.1.2 Types de mouvement de base.....	9
2.1.3 Les organes impliqués dans les mouvements de base	9
2.2 La posture humaine.....	10
2.2.1 Définition.....	10
2.2.2 Conditions du maintien de la posture	11
2.3 La marche humaine.....	11
3 La capture du mouvement humain	11
3.1 Les systèmes de capture de mouvement existants	12
3.1.1 Les systèmes avec marqueurs.....	12
3.1.2 Les systèmes sans marqueurs	14
4 L'animation d'Avatar	14
4.1 Définition d'un Avatar	14
4.2 La réalité virtuelle	15
4.3 L'utilisation d'Avatars dans le monde virtuel	16
4.4 Techniques d'animation dans le monde virtuel	16
4.4.1 Animation à clé.....	16

4.4.2	Animation procédurale	17
4.4.3	L’animation des Avatars dans le monde virtuel	18
4.4.3.1	La chaine cinématique	18
4.4.3.2	La cinématique directe	19
4.4.3.3	La cinématique inverse	20
5	Le capteur de profondeur Kinect de Microsoft.....	22
5.1	Caractéristiques de la Kinect	23
5.2	Fonctionnement de la Kinect	24
5.2.1	Principe de mesure de la profondeur par triangulation.....	25
5.2.2	Extraction du squelette du sujet humain.....	26
6	La fusion de données dans les systèmes multi-capteurs	28
6.1	Les méthodes de fusion de données.....	29
6.1.1	Approche bayésienne.....	29
6.1.2	Le filtre de Kalman.....	29
6.1.3	Les filtres à particules.....	30
6.2	Architectures des systèmes de fusion de données multi capteurs.....	30
6.2.1	Architecture centralisée	30
6.2.2	Architecture distribuée	30
6.2.3	Architecture hiérarchisée	31
6.2.4	Architecture décentralisée	31
7	Conclusion	31
	Chapitre 2 Les systèmes multi-Kinect pour la capture des mouvements	32
1	Introduction.....	33
2	Domaine d’utilisation de la capture de mouvements.....	33
3	L’étude et l’analyse des systèmes similaires existants.....	36
4	L’approche proposée	41
5	Conclusion	42
	Partie II Système Multi-Kinect développé pour la capture de mouvements.....	43
	Chapitre 3 Conception.....	44
1	Introduction.....	45
2	Architecture du système proposé : Présentation globale de la solution.....	45
2.1	Acquisition et prétraitements des données.....	48
2.2	Calibrage.....	48
2.3	Ajustement et fusion des données pour l’animation d’Avatar.....	49

2.4	Sauvegarde et représentation graphique de données squelettiques.....	50
2.5	Animation différée	51
3	Modélisation UML	51
3.1	Diagrammes de cas d'utilisation.....	52
3.2	Diagramme de classes.....	55
3.3	Diagrammes de séquence.....	57
3.4	Diagramme de déploiement	60
4	Installation et configuration matérielle	61
5	Processus de l'animation virtuelle de l'Avatar	63
5.1	Acquisition et prétraitements de données squelettiques brutes.....	63
5.1.1	Extraction du squelette	64
5.1.2	Prétraitement des données squelettiques	65
5.1.2.1	Auto-intersection	65
5.1.2.2	Cinématique directe	67
5.1.2.3	Filtre de lissage	68
5.1.3	Détection de visage.....	68
5.2	Calibrage	70
5.2.1	Représentation d'une articulation	70
5.2.2	Changement de repère	70
5.3	Fusion des données squelettiques et animation de l'Avatar	74
5.3.1	La fusion des données.....	74
5.3.1.1	Acquisition des données et identification des clients	75
5.3.1.2	Sélection du squelette de base ou de référence.....	75
5.3.1.3	La fusion des données squelettiques.....	76
5.3.2	Processus d'animation de l'Avatar	77
5.3.2.1	Transformation des données squelettiques	77
5.3.2.2	Adaptation du modèle 3D de l'Avatar avec la nature du modèle squelettique.....	78
5.3.2.3	L'animation de l'Avatar	79
5.4	Sauvegarde et représentation graphique de données squelettiques.....	80
5.5	L'animation différée	82
6	Conclusion	83
	Chapitre 4 Réalisation et tests.....	84
1	Introduction.....	85

2	Technologies et outils utilisés.....	85
2.1	Environnement matériel.....	85
2.2	L'environnement logiciel.....	86
3	Présentation du système développé	88
3.1	Authentification et inscription	88
3.2	Menu principal	93
3.2.1	Le panneau du profil « Médecin ».....	93
3.2.2	Le panneau « Gestion des patients »	94
3.2.3	Le panneau « Nouveau Examen »	97
3.2.4	Le panneau « Consulter Examen ».....	101
4	Tests et évaluations.....	105
4.1	Configuration du système développé.....	105
4.1.1	Configuration avec un seul capteur	105
4.1.2	Configuration avec deux capteurs	106
4.1.3	Configuration avec trois capteurs	106
4.1.4	Configuration avec quatre capteurs	107
4.1.5	Configuration avec cinq capteurs	107
4.2	Etude expérimentale.....	108
4.2.1	Sauvegarde des données squelettiques	110
4.3	Evaluation quantitative du système développé	111
5	Conclusion	113
	Conclusion Générale.....	114
	Bibliographie	116

Liste des figures

Figure 1.1 : Types du mouvement de base[10].....	9
Figure 1.2 : Quelques postures humaine[16]	10
Figure 1.3: Système de capture de mouvement mécanique[21]	12
Figure 1.4 : Système de capture de mouvement optique[21].....	13
Figure 1.5 : Exemple d’Avatars utilisés [26].....	15
Figure 1.6 : Domaines d'utilisation des Avatars dans la médecine (a), les jeux vidéo (b) et la rééducation (c)[26].....	16
Figure 1.7 : Représentation d’une chaîne cinématique du corps humain et les différents types de liaison [33].	19
Figure 1.8 : Les étapes de contrôle par la technique Cinématique Directe [36].....	20
Figure 1.9 : Les étapes suivies dans la méthode cinématique inverse [36].	21
Figure 1.10 : Exemple de résolution d’un système en utilisant les méthodes analytiques [36].	21
Figure 1.11 : Le capteur Kinect[39].....	23
Figure 1.12 : processus de reconnaissance de posture humaine par la Kinect [41].....	24
Figure 1.13 : Extraction d’une image de profondeur avec la Kinect	25
Figure 1.14: Mesure de la valeur de la disparité d [43]. Mise en évidence de la corrélation entre le motif observé et le motif de référence	25
Figure 1.15: Processus de la reconstruction de la posture humaine proposé par Shotton et al [23].....	26
Figure 2.1 : Exemple de capture de mouvement dans le domaine médical [51]	33
Figure 2.2: Exemple de la capture de mouvement dans le domaine sportif [52].....	34
Figure 2.3 : Exemple de capture de mouvement dans le domaine de la danse [53]	34
Figure 2.4 : Exemple de capture de mouvement dans le domaine du cinéma [54]	35
Figure 2.5: Exemple de capture de mouvement dans le domaine des jeux vidéo [55].....	35
Figure 2.6 : Schéma du système de capture de mouvements pour l’animation d’Avatars [65]	40
Figure 3.1 : Architecture globale du système proposé.....	47
Figure 3.2 : Relation entre les points de vue de deux repères capteurs d’une seule articulation d’un même squelette	49
Figure 3.3 : Processus global de l’animation par la fusion de données	51
Figure 3.4 : Diagramme de cas d’utilisation « Sous système d’acquisition et prétraitements »	53
Figure 3.5 : Diagramme de cas d’utilisation « Sous système d’animation par fusion ».....	55
Figure 3.6 : Diagramme de classes	56
Figure 3.7 : Diagramme de séquence – Ajouter patient.....	57
Figure 3.8 : Diagramme de séquence – Calibrer capteurs	58
Figure 3.9 : Diagramme de séquence – Animation par fusion	59
Figure 3.10 : Diagramme de séquence – Ajouter examen	60
Figure 3.11 : Diagramme de déploiement.....	61
Figure 3.12 : Vue de profil de la position sujet humain/capteur Kinect	62

Figure 3.13 : Schéma global de la configuration matérielle du système proposé.....	63
Figure 3.14 : Représentation du modèle squelettique envoyées par la Kinect	65
Figure 3.15 : Détection et correction de l'auto intersection.....	66
Figure 3.16 : Exemple de cinématique directe appliquée à la main	67
Figure 3.17 : Exemple de suivi du visage fournit par le SDK [67]	69
Figure 3.18 : Représentation d'une articulation dans le repère capteur.....	70
Figure 3.19 : Transformation de repère	71
Figure 3.20 : Calibration des capteurs	73
Figure 3.21 : Approche proposée pour la fusion des données squelettiques	77
Figure 3.22 : Exemple du calcul de la rotation de l'articulation de la main gauche.....	78
Figure 3.23 : Modèle d'Avatar.....	78
Figure 3.24 : Mise en correspondance entre le squelette de la Kinect et le squelette de l'Avatar	79
Figure 3. 25 : Structure JSON des données d'une animation	80
Figure 3.26 : Exemple de sauvegarde des données squelettiques d'une animation.....	81
Figure 3.27 : Exemple de fichier sauvegarder sous-format CSV exploitable par Excel	81
Figure 3.28 : Exemple de représentation 3D de la partie inférieure du cops.....	82
Figure 3.29 : Processus d'animation différée	83
Figure 4.1 : Logo C#.....	86
Figure 4.2 : Logo JSON	87
Figure 4.3 : Logo CSV.....	87
Figure 4.4 : Logo Microsoft Kinect SDK	87
Figure 4.5 : Logo Visual Studio.....	87
Figure 4.6 : Logo Unity	88
Figure 4.7 : Logo Mixamo	88
Figure 4.8 : Logo AUTO DESK.....	88
Figure 4.9 : Interface d'authentification	89
Figure 4.10 : Scène de capture de mouvements « Client ».....	89
Figure 4.11 : Interface d'inscription	90
Figure 4.12 : Interface d'inscription après vérification des données saisies.....	90
Figure 4.13 : Email de confirmation	91
Figure 4.14 : Interface de confirmation d'authentification.....	91
Figure 4.15 : Interface de récupération de l'email pour le changement du mot de passe.....	92
Figure 4.16 : Interface du changement du mot de passe.....	92
Figure 4.17 : Interface du changement du mot de passe.....	92
Figure 4.18 : Interface du menu principal sur la machine serveur.....	93
Figure 4.19 : Panneau Profil	93
Figure 4.20 : Interface de modification du profil.....	94
Figure 4. 21 : Bouton « Déconnexion » et « Quitter ».....	94
Figure 4.22 : Panneau Patients.....	94
Figure 4.23 : Interface gestion des patients	95
Figure 4.24 : Interface ajouter patient.....	95
Figure 4.25 : Interface modifier patient	96
Figure 4.26 : Interface afficher informations patient.....	96

Figure 4.27 :Panneau nouveau examen	97
Figure 4.28: Interface sélection du patient pour un examen	97
Figure 4.29 : Interface de connexion des capteurs Kinect	98
Figure 4.30 : Interfaces des étapes du processus de calibrage des capteurs	99
Figure 4.31 : Scène d’animation 3D en temps réel	99
Figure 4.32 : Fenêtre d’enregistrement des notes	100
Figure 4.33 : Liste des états de suivis des capteurs.....	100
Figure 4.34 : Panneau Consulter Examen.....	101
Figure 4.35 : Interface de sélection du patient pour la consultation d’examen	101
Figure 4.36 : Interface des informations du patient et ses examens	101
Figure 4.37 : Interface consultation d’un examen	102
Figure 4.38 : Interface de sélection du patient pour la consultation d’examen	102
Figure 4.39 : Interface Tableau des données squelettiques 3D d’une animation.....	103
Figure 4.40 : Scène de l’animation différée	103
Figure 4.41 : Interface de sélection des articulations pour la représentation 3D des postures	104
Figure 4.42 : Scène de représentation 3D des postures	104
Figure 4.43 : Scène d’animation squelettique.....	105
Figure 4.44 : Configuration avec un seul capteur	106
Figure 4.45 : Configuration avec deux capteurs	106
Figure 4.46 : Configuration avec trois capteurs.....	107
Figure 4.47 : Configuration avec un quatre capteurs	107
Figure 4.48 : Configuration avec cinq capteurs	108
Figure 4.49 : Expérience de configuration avec cinq capteurs	109
Figure 4.50 : Exemple d’animation d’un mouvement	110
Figure 4.51 : Exemple d’un fichier « JSON » des données d’une animation.....	111
Figure 4.52 : Exemple d’un fichier « CSV » des données d’une animation.....	111
Figure 4.53 : Temps d’exécution de l’algorithme de fusion développé et animation en fonction du nombre des capteurs utilisés	112
Figure 4.54 : Temps de trajet moyen (RTT) des données client en fonction du nombre des capteurs utilisés.....	113

Liste des tableaux

Tableau 2.1: Tableau comparative des travaux sur la capture du mouvement	38
Tableau 4.1 : Caractéristiques des machines utilisées	85
Tableau 4.2 :Matériel informatique utilisé.....	86

Introduction générale

La réalité virtuelle (RV) est un domaine pluridisciplinaire qui a connu des progressions spectaculaires offrant aux utilisateurs des aptitudes d'interactions et d'immersions variées et performantes. Ce domaine qui se trouve à la croisée des chemins des sciences de l'ingénieur et des sciences humaines permet aux utilisateurs de mener une activité sensori-motrice et cognitive dans un environnement virtuel interactif. Des systèmes basés sur ces techniques dans divers domaines se sont réalisés à savoir l'éducation, l'entraînement militaire et la médecine. Dans ce dernier domaine, la RV est devenue un nouvel outil thérapeutique majeur pour l'identification des troubles neurologiques et pour la rééducation des personnes handicapées et la réadaptation locomotrice.

Dans cette optique, des chercheurs ont proposé plusieurs systèmes pour le corps médical, par exemple la capture des mouvements humains d'un patient, par un système de vision 3D dans une séance de rééducation. En outre, la représentation visuelle d'une personne est l'un des principaux problèmes rencontrés dans les environnements virtuels. Elle permet de mettre en corrélation des séquences de mouvements effectués par le patient (le malade) et l'animation de sa représentation 3D dans le monde virtuel.

Pour améliorer le degré de la perception artificielle, le patient doit être libre de se mouvoir dans l'environnement naturel sans contraintes physiques en dupliquant ses mouvements par un Avatar via des techniques d'interaction homme/machine (IHM) qui donnent la sensation du réalisme des mouvements de l'environnement virtuel. En conséquence, l'animation du corps d'un être humain par un Avatar augmente considérablement la liaison entre le monde réel et le monde virtuel et facilite la prise en charge des patients par le médecin.

Le domaine de recherche de vision artificielle s'est largement amélioré avec le développement des capteurs 3D tel que la Kinect de Microsoft qui aide au développement de logiciels de compréhension des mouvements humains et l'animation des Avatars. Ces capteurs Kinect qui sont avant tout peu coûteux, pas trop encombrants et assez précis, permettent l'acquisition d'information physique du corps humain sans interférence avec la scène observée. Ils sont très rentables lors de l'analyse biomécanique du mouvement humain.

Problématique :

Plusieurs secteurs tels que la médecine ont incorporé des techniques de capture des mouvements humains par la vision artificielle. Par exemple, les spécialistes du domaine de la médecine physique et de la réadaptation fonctionnelle (rééducation fonctionnelle) ont trouvé que les outils traditionnels sont généralement coûteux et encombrants. Aussi, l'énorme manque d'interactivité dans les séances de rééducation pousse les patients à manquer d'enthousiasme et à se décourager facilement en raison de la coopération passive des patients avec les thérapeutes. De plus, la complexité d'indentification des anomalies des mouvements humains et plus particulièrement la marche humaine engendre souvent des diagnostics faussés ce qui mène à la non satisfaction des recommandations thérapeutiques.

En outre, le traitement d'intelligence artificielle des données renvoyées par la capture de mouvements permet de dégager des informations robustes et pertinentes qui pourraient être utilisées dans plusieurs domaines tels que l'analyse des mouvements et l'analyse de la marche humaine. Cependant, pour assurer le réalisme, la précision et la robustesse des mouvements capturés, les données squelettiques et biomécaniques extraites des patients doivent être le plus fiable que possible pour que les connaissances et les informations extraites soient pertinentes et plus précises.

La capture des mouvements en utilisant un seul capteur est très limitative dans le cas des systèmes d'analyse des mouvements et de la marche humaine. D'abord, le champ de la capture est trop limité et oblige le sujet humain à se placer face au capteur tout au long de la capture pour bien suivre les différentes parties de son corps et d'extraire au mieux les données squelettiques, ce qui limite sa liberté de se mouvoir dans la scène. De plus, ce type de capture souffre d'un autre problème majeur qui est l'occultation des parties du corps. En Effet, ces inconvénients réduisent fortement la flexibilité de ces systèmes, la robustesse du suivi et la qualité de la capture des mouvements.

Objectifs :

Le but de l'analyse des données biomécaniques dans le domaine de la rééducation fonctionnelle est d'apporter aux membres soignants des connaissances pour l'aide au diagnostic et à l'interprétation clinique. Ce qui leur permet de suivre et de contrôler l'évolution des états de leurs patients et ceci pour assurer le traitement adapté et d'améliorer les recommandations et les choix thérapeutiques et mieux identifier les causes des défauts de la marche. L'analyse des mouvements et de la marche humaine fournit un apport incontestable pour la prise en charge des patients car la déduction des causes probables des anomalies conditionne directement le traitement à réaliser.

C'est dans ce contexte que se situe notre présent travail qui a pour objectif de concevoir et de développer un outil capable de capturer le mouvement du corps humain pour l'animation en temps réel et le contrôle d'un Avatar dans un monde virtuel. À cette fin, nous proposons un système de capture des mouvements humains multi capteurs offrant une couverture complète de la zone de la capture avec un champ de vision de 360° permettant d'aboutir à un suivi complet et robuste des postures et des mouvements exercés par le sujet humain d'une façon réaliste et précise. Cela est garanti par la normalisation des données provenant des différents capteurs et l'élaboration d'une procédure fiable de fusion des données résultantes afin de reproduire correctement le mouvement 3D du sujet humain sur un Avatar dans l'environnement virtuel.

Notre objectif dans ce travail est de concevoir un système de capture de mouvement sous différents points de vue observés chacun par un capteur Kinect et de fusionner l'information de chaque point de vue plutôt que d'utiliser les données d'un seul point de vue. Cette stratégie de redondance de l'information permettra de régler les problèmes d'occultations dus aux données manquantes provenant d'un seul capteur Kinect et offre un espace observé modulable selon les besoins tout en évitant plusieurs mauvaises captures de position de joints du modèle du sujet suivi. De plus, prétraiter localement l'information provenant de chaque capteur Kinect et de ne transférer que ce qui est nécessaire à la fusion. Cependant cette approche n'est fonctionnelle que si les données provenant des différentes sources sont représentées dans un seul repère global.

Un autre objectif est d'élaborer en premier lieu, une méthode permettant de calibrer les différents capteurs utilisés dans la capture de mouvements dans le but de garantir les

déplacements entre les différents points de vue des caméras couvrant l'espace de travail et d'assurer l'homogénéité des données fournies par les capteurs lors de la fusion. Puis, en second lieu est de concevoir une procédure fiable de fusion des données 3D en temps réel afin de combiner les multiples données d'une manière appropriée et d'améliorer la qualité et la précision de la capture de mouvement.

Du coup, capturer les mouvements exercés par un sujet humain dans un contexte réel, pour les reconstruire dans un environnement virtuel à travers l'animation d'un Avatar par des mouvements réalistes précis et fluides, constitue l'objectif principal de ce travail.

Plan du mémoire

Notre mémoire est organisé comme suit : Le premier chapitre abordera une revue de littérature sur les systèmes et les méthodes liés à la thématique abordée dans ce travail. Dans le deuxième chapitre seront présentés les différents domaines d'utilisation de la capture des mouvements ainsi que les problèmes et les solutions proposées dans la littérature.

Nous consacrons l'intégralité du troisième chapitre à la conception du système proposé et détaillons chaque élément qui le compose, tout en discutant de la méthodologie proposée afin de résoudre le problème de capture de mouvements multi-vues en temps réel pour un sujet progressant dans l'espace du travail.

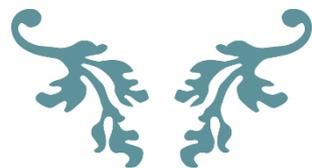
Le quatrième chapitre, quant à lui, couvrira les aspects techniques de l'implémentation de notre système ainsi que la discussion et l'évaluation des résultats expérimentaux effectués.

Nous achevons ce mémoire avec une conclusion générale où nous faisons une synthèse du travail effectué tout en fournissant une liste de perspectives, en vue d'améliorer le système proposé.



Partie I

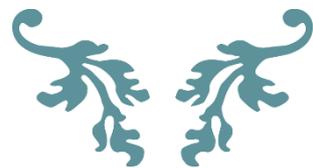
Systemes de capture de mouvement





Chapitre 1

**Systeme de capture de
mouvement : Etat de l'art**



1 Introduction

L'analyse du mouvement humain intéresse de nombreux domaines d'application. Dans le domaine de la médecine par exemple, nous pouvons citer la détection des anomalies de la marche chez un patient atteint par des pathologies neurologiques ou autres. La proposition de solutions à l'intégration des handicapés dans la société moderne, en utilisant des outils de diagnostic optique et d'évaluation du comportement des patients qui sont généralement coûteux, est plus que nécessaire.

De ce fait, la capture du mouvement humain est devenue un moyen de traitement très reconnu dans le domaine médical et intéresse un grand nombre de chercheurs. Cette technique permet d'enregistrer les positions et les rotations d'un sujet humain en temps réel, afin de les reproduire, les analyser, et les comparer par rapport à celles d'un sujet normal.

Dans ce chapitre, nous allons tout d'abord définir ce qu'est un mouvement, puis nous définirons les concepts relatifs à la posture humaine, à la marche humaine et à l'analyse du mouvement humain.

2 Analyse du mouvement humain

Le mouvement du corps humain contient des subtilités qui doivent être bien analysées. Cette analyse vise à rassembler des informations quantitatives sur les mécanismes du système musculo-squelettique lors de l'exécution d'une tâche motrice par observation attentive du mouvement humain dans le but de le comprendre, d'estimer et d'évaluer sa qualité [1].

Les auteurs *Shun et al.* ont démontré dans [2] la faisabilité de la reconnaissance des états émotionnels à partir de l'analyse des mouvements humains en s'intéressant aux allures que prend un promeneur qui pourraient refléter son état émotionnel. Ils ont utilisé un système de capture de mouvements Multi-Kinect pour enregistrer les allures des personnes à trois états émotionnels (neutre, colère et bonheur) induits, et sur lesquelles ils ont effectué une classification des données afin de décider l'état du promeneur.

De ce fait, la capture de mouvement est devenue une technique indispensable, permettant d'enregistrer les positions et les rotations d'objets ou des humains, pour en contrôler une contrepartie virtuelle sur ordinateur [3].

2.1 Mouvement humain

2.1.1 Définitions

Comprendre le mouvement humain intéresse les chercheurs de nombreuses disciplines scientifiques que ce soit la physiologie, la kinésiologie, la biomécanique, l'anatomie fonctionnelle, les sciences du sport ou l'informatique. Cela mène à plusieurs définitions du terme *Mouvement*.

En physiologie et notamment en anatomie fonctionnelle, descriptive ou mécanique, le mouvement est un changement de position dans l'espace du corps ou d'une partie du corps d'un être vivant [4].

Un mouvement, dans le domaine de la mécanique (physique), est le déplacement d'un corps par rapport à un point fixe de l'espace nommé référentiel, à un moment déterminé. Le mouvement est caractérisé par sa trajectoire et l'évolution de sa vitesse [5].

Il est aussi défini par la variation de la position d'un point, d'un solide ou d'un système, étudié dans un référentiel donné, en fonction du temps. Un mouvement est décrit grâce à une vitesse et à une trajectoire. En fonction de ces derniers, il existe plusieurs types de mouvement : un mouvement rectiligne uniforme accéléré , un mouvement circulaire ralenti ou encore décéléré et un mouvement curviligne [6].

En kinésiologie, les mouvements sont contraints par les caractéristiques du squelette et des muscles. Le cerveau joue un rôle important pour le contrôle de certains mouvements, mais de nombreux mouvements sont générés par des systèmes réflexes, pour le contrôle de l'équilibre notamment, en lien avec le système vestibulaire [4].

Dans le domaine de l'anatomie, le mouvement est considéré comme un système organisé. Le niveau de départ de l'ordre moteur est donc cérébral et cortical. Dans son déroulement, le mouvement est une suite organisée d'actions musculaires successives recrutant à la fois, les muscles synergiques (agonistes et antagonistes), dans chaque segment du membre mobilisé [7].

Du point de vue fonctionnel, *Dalley et Moore* ont mentionné dans [8] que les mouvements sont associés à des articulations où au moins deux os qui s'articulent avec un autre os.

En conclusion, le mouvement humain représente une suite de postures que prend le corps humain en un laps de temps [9].

2.1.2 Types de mouvement de base

Tous les mouvements du corps humain résultent de la combinaison de mouvements simples (mouvements de base). Quatre types de mouvement ont été présentés dans [10] comme le montre la figure 1.1.

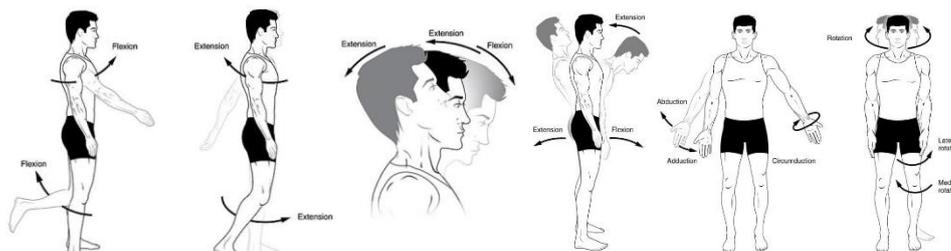


Figure 1.1 : Types du mouvement de base[10]

a) **Le mouvement de flexion** : tout mouvement permettant de replier une partie d'un membre sur un autre ou une partie du corps sur une autre est appelé mouvement de flexion.

b) **Le mouvement d'extension** : tout mouvement permettant de placer deux parties d'un membre ou du corps dans le prolongement l'une de l'autre est appelé mouvement d'extension.

c) **Le mouvement de rotation** : tout mouvement permettant de faire pivoter une partie du corps ou d'un membre par rapport à une autre est un mouvement de rotation.

d) **Les mouvements d'abduction et d'adduction** : l'abduction est un mouvement qui écarte un membre, de l'axe médian du corps dans un plan frontal. Il s'oppose ainsi à l'adduction.

2.1.3 Les organes impliqués dans les mouvements de base

Dans tout mouvement simple interviennent plusieurs organes : muscles, os et nerfs [10].

a) **Les muscles** : ils sont fixés aux os de squelette par des tendons. Ils interviennent dans le mouvement de façon active.

b) **Les os** : les os du squelette étant rigides, ils ne peuvent pas se déplacer, l'un par rapport à l'autre, qu'au niveau de leur zone de contact appelée articulation.

c) **Les nerfs** : aucun mouvement ne serait possible sans l'intervention du nerf qui met en relation un muscle avec un centre nerveux en transmettant un message nerveux.

2.2 La posture humaine

2.2.1 Définition

Au sens premier, le terme « posture » indique une attitude, une façon de se tenir ou de placer son corps ou ses membres (se tenir debout, penché, raide, détendu, etc.) [11]. Selon le dictionnaire Larousse, il s'agit de la position du corps ou d'une de ses parties dans l'espace. Le système postural est responsable de la régulation de l'équilibre du corps. La posture peut donc se définir, conceptuellement, comme « une façon d'occuper une position » [12].

Au sens physique, le concept de posture est associé à la corrélation entre les membres et le tronc, et aux positions des articulations. Dans un contexte plus exact, la posture est la position du corps humain. Ventrale, dorsale, latérale et couché sont quelques-uns des noms techniques que reçoivent certaines positions du corps [13].

Une posture est définie aussi par une position particulière des segments corporels les uns par rapport aux autres à un instant donné [14] [15]. Il existe un grand nombre de postures pouvant être adoptées par un être humain. Parmi elles, la station debout érigée constitue la posture de référence. La posture idéale est un alignement vertical dans lequel notre système musculo-scoliotique dépense le moins d'efforts pour se tenir droit.

En informatique pour pouvoir modéliser la posture du corps humain, il faut assigner au corps un modèle qui le caractérise, pour décrire le plus fidèlement possible les postures que prendra le corps. Parmi les représentations du corps que nous retrouvons le plus, nous avons les modèles squelettiques et les modèles basés sur la forme de la silhouette [16] comme le montre la figure 1.2.

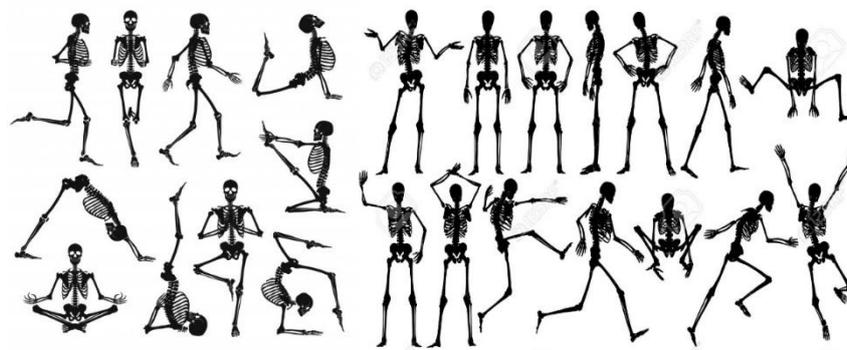


Figure 1.2 : Quelques postures humaine[16]

2.2.2 Conditions du maintien de la posture

A. *Claire et al.* ont défini dans [17] les conditions du maintien de la posture selon deux points de vue :

1. **Du point de vue mécanique** : Trois conditions doivent être réunies pour tenir une posture sur terre :
 - a) Reconnaître la verticale gravitaire (oreille interne) et, ainsi, la direction des forces exercées par l'attraction terrestre ;
 - b) Stabiliser le centre de gravité de l'ensemble du corps ou le maintenir placé au-dessus de la base d'appui (à la verticale de la base d'appui) ;
 - c) Disposer d'un polygone de sustentation ou base d'appui (indépendamment de la disposition des segments corporels les uns par rapport aux autres).
2. **Du point de vue physiologique** : Trois conditions doivent pouvoir être assurées :
 - a) Pouvoir reconnaître la verticale en rapport avec la fonction de l'oreille interne ;
 - b) Pouvoir stabiliser chaque segment à un instant donné (appui, forces de maintien) et composer la stabilité de l'ensemble;
 - c) Disposer des appuis suffisamment stables et rigides pour contrôler (équilibrer) les forces en présence des forces intérieures et extérieures.

2.3 La marche humaine

La marche normale humaine peut être définie comme un phénomène complexe qui implique la coordination des mouvements de rotation des segments corporels pour maintenir l'équilibre du corps humain pendant son déplacement vers l'avant. Elle s'acquiert à travers un processus d'apprentissage et les patrons caractéristiques de la marche dite normale prennent plusieurs années à se développer. Dans ce processus, un individu imprime des caractéristiques distinctives sur son pattern de marche avec implicitement la volonté de se déplacer de façon économique [18].

3 La capture du mouvement humain

La capture de mouvement (Motion Capture ou Mocap) est un sujet en plein essor qui intéresse plusieurs secteurs à savoir : les jeux vidéo, le sport, la médecine et la cinématographie [19]. C'est une technique qui permet d'enregistrer les positions et rotations d'un sujet humain en temps réel, afin de les reproduire ou bien de les analyser [20]. Elle peut être considérée

comme un processus d'enregistrement du mouvement réel d'un sujet humain sous forme de séquences de coordonnées cartésiennes dans l'espace 3D. Il s'agit de capturer le plus précisément les mouvements d'un sujet humain afin de les numériser et de les utiliser dans un contexte particulier [20].

3.1 Les systèmes de capture de mouvement existants

Différents systèmes de capture existent, ils diffèrent essentiellement par leur technologie et sont de nos jours fréquemment utilisés en production pour l'animation de personnages et l'analyse des mouvements. Deux types de systèmes de capture de mouvement existent : ceux avec marqueurs et ceux sans.

3.1.1 Les systèmes avec marqueurs

Ces systèmes utilisent des marqueurs qui seront fixés sur les articulations du sujet humain pour être capturés par des caméras dispersées. Nous présentons quatre types de ces systèmes.

a. Les systèmes mécaniques : ils fonctionnent grâce à un exosquelette, une combinaison comprenant des dispositifs tels que des accéléromètres, situés aux articulations fournissant des informations sur leur position et leur orientation [21] comme l'illustre la figure 1.3.

Ces systèmes sont relativement peu utilisés malgré l'avantage de leur précision, il n'y a aucun risque de perte durant le mouvement et cela assure une certaine stabilité à l'enregistrement.

Malgré leur fiabilité et précision des données, ces systèmes ont l'inconvénient d'être encombrants et empêchent le corps humain à faire un mouvement plus correct.



Figure 1.3: Système de capture de mouvement mécanique[21]

b. Les systèmes magnétiques : ils consistent à utiliser un ensemble de récepteurs magnétiques situés sur les articulations du sujet, chacun étant connecté par un câble à un ordinateur. Un émetteur magnétique permet de mesurer la position des récepteurs entre eux [21]. Ces systèmes sont précis et rapides pour des captures de mouvements simples, mais ont le désavantage d'être limités par l'utilisation des câbles.

c. Les systèmes inertiels : ils utilisent des centrales inertielles ou gyroscopiques permettant l'acquisition de l'attitude du corps humain en temps réel, par rotation de ses membres, en supprimant les contraintes imposées par les technologies optiques. En effet, ces systèmes continuent à fonctionner même cachés sous les habits ou accessoires. Par contre, ceux-ci n'enregistrent pas leurs positions, qui doivent être réalisées avec d'autres capteurs [22]. Ces systèmes ont l'avantage d'être légers et portables et de ne pas nécessiter de caméras et ne souffrent pas des problèmes d'occultations de marqueurs, par contre ils peuvent cependant perdre de la précision sur de longues captures, ils sont moins précis.

d. Les systèmes optiques : ils utilisent des marqueurs réfléchissants et des caméras digitales infrarouges disposées de manière à entourer une zone de capture comme le montre la figure 1.4. La position des marqueurs dans l'espace est calculée par triangulation, connaissant les paramètres intrinsèques de chaque caméra ainsi que la position des marqueurs dans chaque vue de caméra [21]. Ils sont les plus populaires malgré leur coût car ils fournissent des données plus précises.

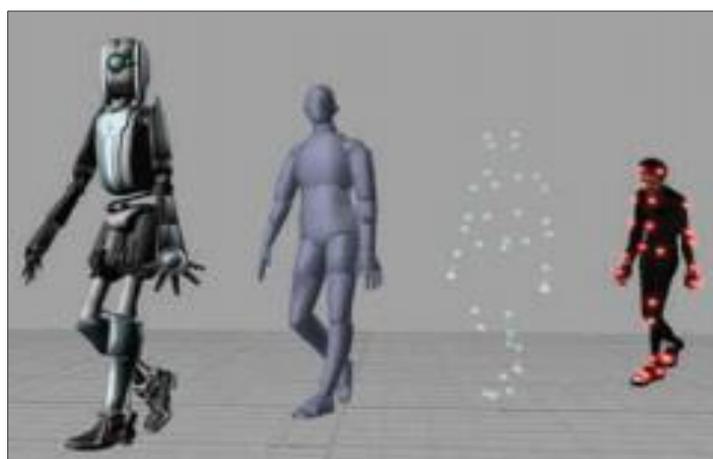


Figure 1.4 : Système de capture de mouvement optique[21]

Les sujets humains doivent porter des équipements qui encombrant leur activité et ces équipements sont très lourds pour être utilisés sur des malades neuromusculaires.

3.1.2 Les systèmes sans marqueurs

Les techniques de capture de mouvement sans marqueur (habituellement appelés Markerless Motion Capture) sont souvent basées sur des caméras. Aussi, depuis la venue de la caméra 3D Kinect de Microsoft, plusieurs solutions sont basées sur cette dernière.

a. Système à base de cameras RGB

Une technique largement utilisée avec des caméras RGB demande un calibrage initiale, permettant de détecter l'arrière-plan. Ensuite, cet arrière-plan peut être soustrait des images nouvellement capturées afin d'isoler le sujet capturé. Cette technique a été jumelée à la capture sonore de chaque caméra utilisée pour la synchronisation des images, le suivi de trait et la densité des détails de l'arrière-plan afin de stabiliser les résultats obtenus *Hasler et al.* Le squelette du sujet capturé est ensuite extrait des images capturées par la caméra RGB et on peut en extraire les positions de chaque joint dans l'espace [19].

b. Système à base de camera de profondeur

Les caméras de profondeurs sont devenues un outil essentiel à la capture des mouvements sans marqueurs, non seulement pour leur faible coût mais aussi pour les Middleware et kits de développement avec ces dernières qui fournissent un squelette complet du sujet humain [23].

Ils utilisent un modèle articulé du corps humain afin de préciser la capture de mouvement. Cette technique offre une façon facile d'extraire le squelette des images de profondeur fournies par la caméra 3D [19]. Par contre, la limite de la précision de la capture est toujours déterminée par la résolution en profondeur de la caméra utilisée.

4 L'animation d'Avatar

Nous commençons cette partie avec une définition de l'Avatar et de la réalité virtuelle. Puis nous nous focalisons sur les différentes techniques d'animation d'Avatars, sur les applications réalisées dans cette optique ainsi qu'à leurs domaines d'utilisation dans le monde virtuel.

4.1 Définition d'un Avatar

Le mot « Avatar » vient du mot *avatara* en sanskrit, qui veut dire « descente », en parlant d'une incarnation ou une manifestation corporelle d'un être immortel dans l'hindouisme. Selon

leurs traditions, les Avatars sont les incarnations successives du dieu Vishnou dans notre monde terrestre [24].

Nous définissons l'Avatar comme étant une représentation contrôlée par l'être humain. D'après *Bailenson* [24], toute forme de représentation qui marque l'entité d'un utilisateur dans les médias sociaux, jeux vidéo, ou environnement virtuel peut être considérée comme un Avatar. un nom, une voix, une photo...etc. Tout peut servir d'Avatar d'utilisateur, même s'il ne ressemble pas nécessairement à l'utilisateur (voir la figure 1.5). Aussi selon *Bailenson* [25], dans un contexte d'interaction homme-machine, les êtres humains contrôlent les Avatars à travers des actions en temps réel, ici les Avatars sont des représentations numériques de personnes réelles. Autrement dit, nous pouvons les considérer comme des interfaces entre la réalité et le virtuel permettant à l'utilisateur une incarnation immersive dans un environnement virtuel.



Figure 1.5 : Exemple d'Avatars utilisés [26]

4.2 La réalité virtuelle

Selon Jean Segura dans [26], spécialiste de l'imagerie de synthèse et des usages du virtuel, la réalité virtuelle peut se définir comme "l'ensemble des techniques et des systèmes qui procurent à l'homme le sentiment de pénétrer dans des univers synthétiques créés sur internet". La réalité virtuelle donne la possibilité d'effectuer en temps réel un certain nombre d'actions définies par un ou plusieurs programmes informatiques et d'éprouver un certain nombre de sensations visuelles, auditives ou haptiques.

La réalité virtuelle fait référence aux environnements d'apprentissage virtuels qui motivent la possession réelle d'un être humain ou d'une chose et les voies sensorielles substantielles. Elle utilise les interactions de l'animation 3D que les utilisateurs peuvent expérimenter avec le clavier et la souris ou avec un casque conçu pour immerger les utilisateurs dans des environnements virtuels réalistes. En outre, certaines applications récentes permettent également aux utilisateurs de sentir et d'interagir avec des objets via une rétroaction forcée [27].

4.3 L'utilisation d'Avatars dans le monde virtuel

Les Avatars sont principalement les habitants des mondes virtuels, l'outil avec lequel des millions d'utilisateurs interagissent, collaborent et forment des relations entre eux dans des environnements en ligne.

Le but des mondes virtuels est de simuler une réalité alternative au sein de laquelle les utilisateurs peuvent s'adonner à des activités qu'ils ne pourraient ou ne voudraient pas accomplir dans le monde réel.

Les mondes virtuels ont alors permis aux gens de vivre une vie sociale humaine dans un environnement où ils interagissent avec le corps (Avatar) qu'ils désirent ou choisissent. Les Avatars 3D ont prouvé leurs utilités dans beaucoup de domaines tels que : la clientèle automatisée, la rééducation dans le domaine de santé, les applications militaires, les jeux vidéo, et le cinéma (voir la figure 1.6).



Figure 1.6 : Domaines d'utilisation des Avatars dans la médecine (a), les jeux vidéo (b) et la rééducation (c)[26]

4.4 Techniques d'animation dans le monde virtuel

L'animation d'humains virtuels est un problème récurrent en synthèse d'images. Deux approches ont été développées au cours de ces dernières années. La première vise à améliorer les techniques de manipulation d'images clés pour augmenter le contrôle de l'infographiste sur l'animation. La seconde approche se focalise sur la génération automatique de mouvements à partir de la simulation des forces animant un corps humain.

4.4.1 Animation à clé

C'est une technique d'animation traditionnelle apparue initialement dans les dessins animés. Un animateur expérimenté dessine les images clés qui déterminent les étapes principales de la décomposition du mouvement, les autres dessinateurs dessinent les images restantes pour compléter le mouvement [28].

Elle consiste à placer les postures du personnage dans une scène, et à laisser le logiciel calculer le mouvement effectué pour passer de l'une à l'autre. Ce calcul, permettant de recomposer un mouvement à partir de deux positions s'appelle l'interpolation [29].

Dans le monde virtuel nous utilisons des images 3D constituées de voxels ; le modèle voxel est la représentation du pixel en 3D. Il consiste à stocker une information colorimétrique avec ses coordonnées spatiales [30].

V. Chandru *et al.* ont proposé dans [31] une méthode d'animation qui ressemble à l'animation traditionnelle par images clés mais dans un monde 3D. Ils considèrent que l'animation de volume où du monde 3D et ses composants sont représentés sous la forme d'un modèle voxel.

La technique de génération de mouvement par images clés présente les avantages suivants :

- Peut être utilisée avec toutes les représentations du squelette.
- Plusieurs types d'interpolation sont utilisées.
- La qualité du mouvement résultant est très dépendante du choix des postures clés.

Mais elle est limitée par les inconvénients suivants :

- Technique très rudimentaire, car l'algorithme d'interpolation peut provoquer des effets indésirables.
- Le problème se complique lorsque les deux dessins ne sont pas d'une forme proche (n'ont pas le même nombre de sommets par exemple).

4.4.2 Animation procédurale

L'animation procédurale est une animation d'objets virtuels par génération en temps réel de mouvements, selon un ensemble de règles procédurales. Le mouvement est décrit algorithmiquement, car il se base sur les algorithmes écrits pour la génération automatique des mouvements et souvent à l'aide des lois physiques [28]. L'animation procédurale est généralement utilisée dans les films d'animation ou les jeux vidéo.

L'animation procédurale est une animation basée principalement sur des algorithmes, à partir d'équations mathématiques et de lois physiques, où un ensemble de données est attribué à l'objet numérique à animer. Ce type d'animation permet de simuler un comportement réel [29].

La technique de génération de mouvement procédurale présente les avantages suivants :

- Pas de gestes indésirables, grâce à l'utilisation des règles mathématiques.
- Adéquate pour la production des animations interactives.

- Lorsqu'on a besoin des nouveaux mouvements, il suffit juste de décrire un nouvel algorithme (procédure).

Mais elle est limitée par les inconvénients suivants à savoir :

- Elle nécessite une compréhension approfondie des paramètres pour générer les mouvements désirés.
- Les animations sont rarement réalistes, car elle ne peut reproduire des mouvements d'humains subtils (comme les expressions faciales ou les mouvements des mains).

4.4.3 L'animation des Avatars dans le monde virtuel

Un modèle 3D peut être muni d'un squelette d'animation et être animé à travers ce dernier. Un squelette d'animation est constitué d'une série d'articulations et d'arcs.

L'état d'un squelette d'animation à un moment précis est appelé une pose. La pose d'un squelette d'animation est définie par la transformation propre à chaque articulation à un moment précis. Il existe plusieurs techniques et méthodes d'animation d'Avatars :

4.4.3.1 La chaîne cinématique

Le mouvement de marche pourrait se résumer à la mobilisation du squelette humain par des actionneurs que sont les muscles squelettiques. Une chaîne cinématique signifie un modèle mathématique des systèmes mécaniques dans lequel un ensemble de solides indéformables sont connectés entre eux par des articulations [32]. Ou encore, une chaîne cinématique, est une représentation hiérarchique, utilisée en robotique. Elle est composée d'un ensemble de liens rigides, qui sont reliés entre eux par des articulations, ces articulations qui permettent de faire bouger chaque lien par rapport à un autre.

Il existe plusieurs types d'articulations, dans le modèle du corps humain représentés dans la figure 1.7, constituant une chaîne cinématique où nous retrouvons des liaisons de type pivot au niveau des coudes et des genoux ainsi que des liaisons sphériques ou rotule au niveau des chevilles, des hanches, des épaules, des poignets et du cou.

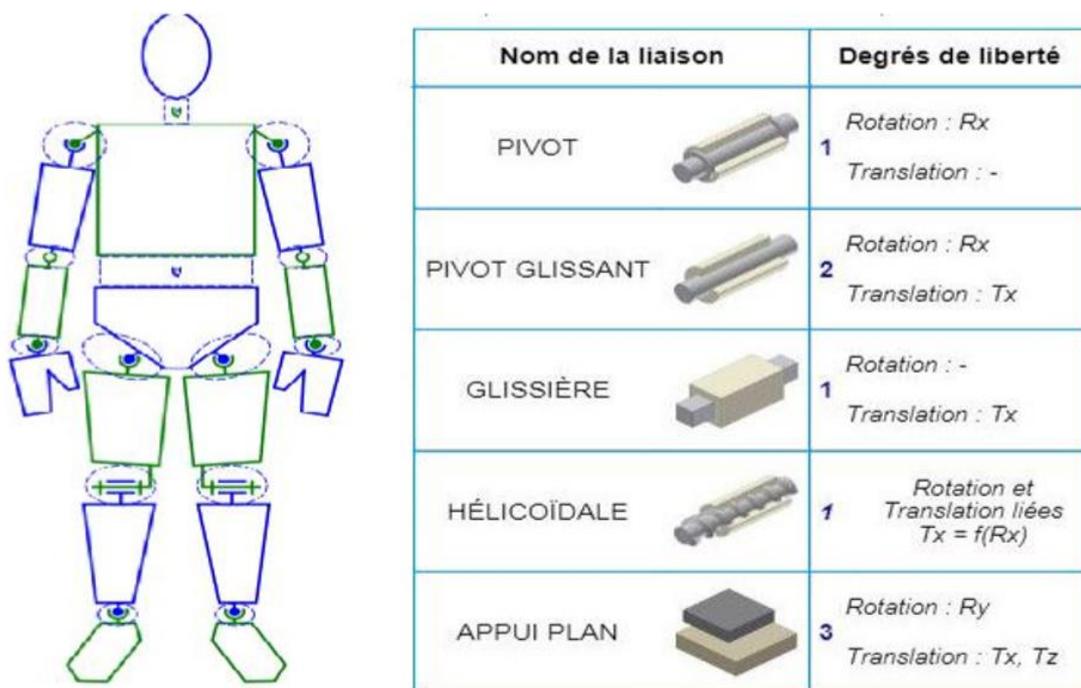


Figure 1.7 : Représentation d'une chaîne cinématique du corps humain et les différents types de liaison [33].

4.4.3.2 La cinématique directe

La cinématique directe est une technique représentée par une chaîne cinématique qui consiste à donner une pose à un objet ou à un corps humain articulé en modifiant uniquement les paramètres de ses articulations. Le mouvement du modèle articulé ou d'une partie de ce modèle peut être considéré comme la combinaison des effets résultants des transformations portées à ces composantes [34].

Le contrôle des mouvements par la cinématique directe est spécifié explicitement par l'animateur en fixant les paramètres de chaque articulation T_i . La position et l'orientation de l'effecteur final M_{Ef} est défini par l'équation suivante [35] :

$$M_{Ef} = f(T_1, T_2, \dots, T_n) = f(T) \quad (1.1)$$

La figure 1.8 montre les étapes de la cinématique directe pour placer la main d'un Avatar articulé dans une position voulue. Ceci dit, en modifiant les articulations de l'épaule, du coude puis du poignet afin de les placer dans des angles permettant d'obtenir la pose voulue. Cette technique est très utilisée dans le domaine d'animation d'Avatars. Elle est très précise car nous contrôlons directement la rotation et la translation des articulations mais la génération d'un mouvement complexe pourrait quand même être un processus laborieux [36].

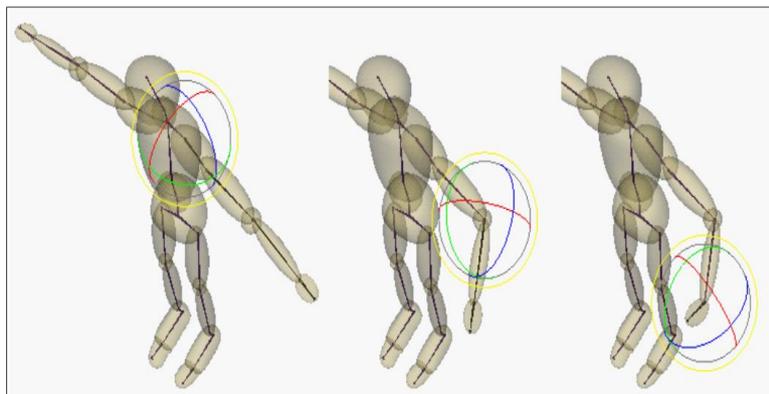


Figure 1.8 : Les étapes de contrôle par la technique Cinématique Directe [36].

- Avantage : La cinématique directe permet de faire des mouvements d'arcs précis et bien définis, puisqu'on contrôle directement les rotations des articulations.
- Inconvénient : Générer et animer des poses complexes peut être un processus laborieux étant donné l'interdépendance entre les éléments du squelette.

4.4.3.3 La cinématique inverse

C'est une technique utilisée dans de nombreux domaines, dont l'animation d'Avatars. Elle est utilisée pour estimer la rotation des articulations, selon la position de l'effecteur final dans une chaîne cinématique. La cinématique inverse s'oppose à la cinématique directe. L'utilisateur définit dans la cinématique inverse la position et l'orientation finale désirée de l'effecteur (M_{Ef}). Après cela, les rotations et les positions d'articulations requises pour atteindre la configuration voulue sont calculées par des méthodes analytique au cas où le mécanisme est simple ou par des méthodes itératives dans le cas où le mécanisme est compliqué [36]. Cela se traduit mathématiquement par la fonction inverse de l'équation de la cinématique directe.

$$T = f^{-1}(M_{Ef}) \quad (1.2)$$

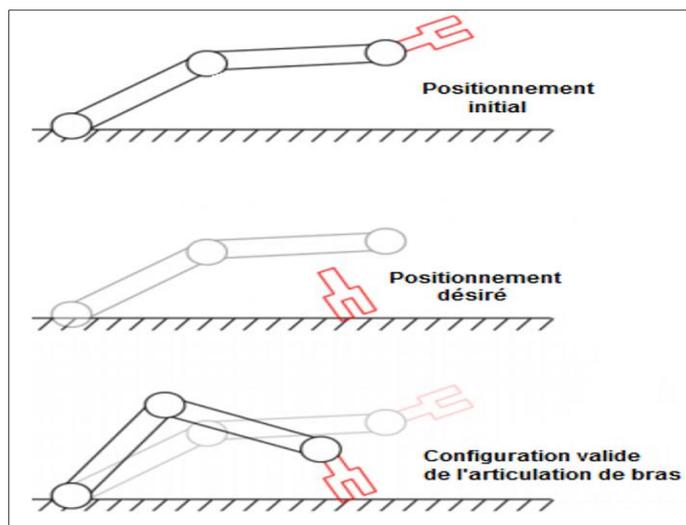


Figure 1.9 : Les étapes suivies dans la méthode cinématique inverse [36].

Il existe plusieurs méthodes pour la résolution de l'équation de la cinématique inverse, elles visent à trouver une configuration d'articulations permettant d'atteindre le positionnement voulu :

❖ Méthodes Analytiques

Les méthodes analytiques consistent principalement à prendre un système et le modéliser à l'aide d'un système d'équations. Nous résolvons alors ce système pour obtenir la configuration souhaitée [36]. La figure 1.10 donne un exemple de résolution d'un système en utilisant les méthodes analytiques.

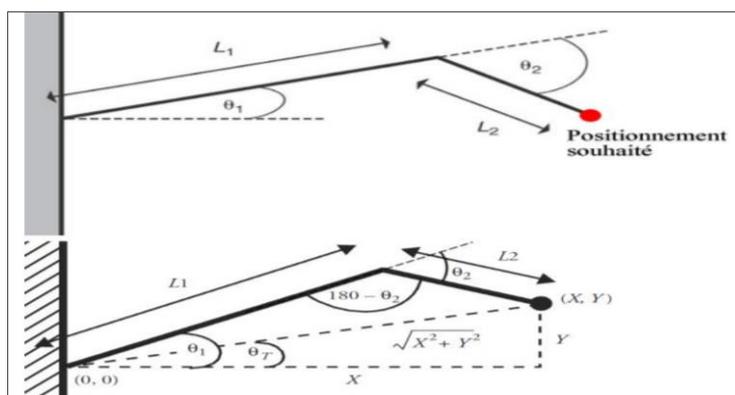


Figure 1.10 : Exemple de résolution d'un système en utilisant les méthodes analytiques [36].

Après nous faisons la génération des équations pour trouver nos angles :

Par définition nous avons :
$$\cos(\theta_T) = \frac{X}{\sqrt{X^2+Y^2}} \quad (1.3)$$

$$\text{Alors : } \theta_T = \text{acos} \left(\frac{X}{\sqrt{X^2+Y^2}} \right) \quad (1.4)$$

$$\text{En utilisant la loi de cosinus : } \cos(\theta_1 - \theta_T) = \frac{L_1^2 + X^2 + Y^2 - L_2^2}{2 \cdot L_1 \cdot \sqrt{X^2+Y^2}} \quad (1.5)$$

$$\text{En remplaçant : } \theta_1 = \text{acos} \left(\frac{L_1^2 + X^2 + Y^2 - L_2^2}{2 \cdot L_1 \cdot \sqrt{X^2+Y^2}} \right) + \theta_T \quad (1.6)$$

$$\text{La loi de cosinus aussi dit : } \cos(180 - \theta_2) = \frac{L_1^2 + L_2^2 - (X^2 + Y^2)}{2 \cdot L_1 \cdot L_2} \quad (1.7)$$

$$\text{Nous obtenons alors : } \theta_2 = \text{acos} \left(\left(\frac{L_1^2 + L_2^2 - (X^2 + Y^2)}{2 \cdot L_1 \cdot L_2} \right) \right) \quad (1.8)$$

Les méthodes analytiques permettent de trouver des résultats pour des systèmes simples, Cependant, dans le cas d'un système plus complexe (pour un plus grand nombre d'articulations) où dans le cas où on souhaite fournir une méthode fonctionnant pour tous les systèmes, la méthode analytique n'est pas adaptée. Dans ce cas il faut avoir recours à des solutions numériques itératives [34].

❖ Méthodes itératives

Ce sont des méthodes qui utilisent une séquence d'étapes qui conduisent à une meilleure solution incrémentale pour les angles des articulations. Elles sont basées sur le principe de minimiser au mieux la différence entre la position actuelle et celle souhaitée de l'effecteur final [37]. Parmi ces méthodes, la méthode du Jacobien inverse, la méthode basée sur l'optimisation, la méthode basée sur le gradient, la méthode de la Descente de Coordonnées Cycliques (CCD) et la méthode de programmation génétique. Certes, auparavant, ces méthodes étaient lentes à s'exécuter, puisqu'elles s'opéraient en des temps de calcul importants, mais vue l'augmentation de la puissance des ordinateurs elles deviennent très recommandées.

5 Le capteur de profondeur Kinect de Microsoft

La Kinect est un capteur de distance, dans la droite lignée des caméras actives à temps de vol. Son exploitation pour la capture de mouvements humains a été initiée par la firme américaine Microsoft [38].

La Kinect de Microsoft est un dispositif rassemblant différents capteurs dont plusieurs microphones, une caméra couleur et une caméra de profondeur. Bien que la Kinect soit dédiée en premier lieu au marché du divertissement, il est possible d'interfacer ses différents capteurs

(caméra RGB, caméra de distance, microphones) directement et donc de l'utiliser pour des projets de recherche et d'ingénierie en adaptant et en améliorant ses capacités. Son arrivée a provoqué un impact important dans le domaine de la recherche en vision artificielle, médicale et robotique [39].

5.1 Caractéristiques de la Kinect

- a. Capteur de profondeur (Technologie infrarouge (IR) contenant une caméra infrarouge et un projecteur infrarouge) comme le montre la figure (Figure 1.11).
- b. Caméra RGB
- c. Microphones
- d. Moteur d'inclinaison

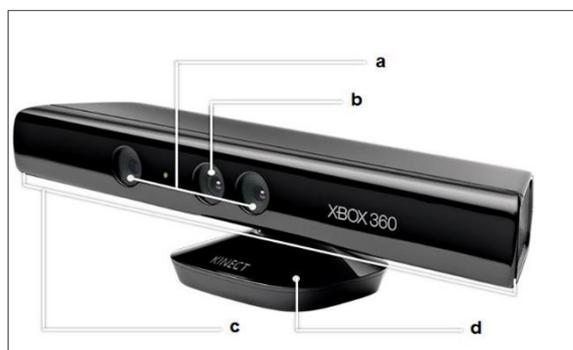


Figure 1.11 : Le capteur Kinect[39]

Capteur :

- Lentilles détectant la couleur et la profondeur
- Micro à reconnaissance vocale
- Capteur motorisé pour suivre les déplacements

Champ de vision :

- Champ de vision horizontal : 57 degrés
- Champ de vision vertical : 43 degrés
- Marge de déplacement du capteur : ± 27 degrés
- Portée du capteur : 1,2 m – 3,5 m

Flux de données :

- 320×240 en couleur 16 bits à 30 frames par seconde
- 640×480 en couleur 32 bits à 30 frames par seconde
- Audio 16 bits à 16 kHz

Système de reconnaissance physique :

- Jusqu'à 6 personnes dont seulement 2 joueurs actifs à la fois

- 20 articulations par squelette
- Application des mouvements des joueurs sur leurs Avatars Xbox Live

Caractéristique audio :

- *Suppression de l'écho*
- *Reconnaissance vocale multilingue*

5.2 Fonctionnement de la Kinect

Selon D. Ménard dans [19], il existe deux grands logiciels dans le monde des caméras 3D : Microsoft Kinect SDK Microsoft Corp et OpenNI (Open Natural Interface) Prime Sense. Nous allons expliquer dans ce qui suit le principe de mesure de la profondeur utilisant le SDK de Microsoft et le processus de la modélisation squelettique de la posture humaine à partir du capteur Kinect.

La Kinect est basé sur une technologie logicielle développée et sur un capteur de profondeur spécifique qui interprète les informations sur la scène 3D obtenue quelles que soit les conditions de luminosité à travers une lumière infrarouge qui n'est nullement gênante pour l'homme.

R. Lun et W. Zhao dans [40] ont divisé le processus de modélisation squelettique de la posture humaine à partir du capteur Kinect en plusieurs étapes comme l'indique la figure 1.12.

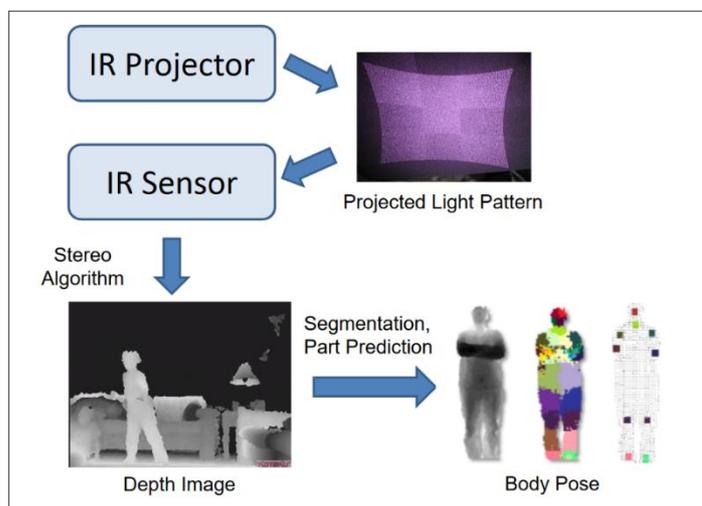


Figure 1.12 : processus de reconnaissance de posture humaine par la Kinect [41]

La Kinect projette un ensemble de rayons de lumière infrarouge structurée en continu grâce à l'émetteur infrarouge et perçoit leurs images sur le plan image du capteur infrarouge, ce qui nous permet de calculer la disparité et mesurer la profondeur en utilisant le principe de triangulation.

5.2.1 Principe de mesure de la profondeur par triangulation

Afin de créer un modèle constant de mouchetures projetées sur la scène, l'émetteur laser de la Kinect émet un faisceau de lumière infrarouge unique qui est subdivisé en faisceaux multiples par un réseau de diffraction.

Le nuage de points projeté sur un objet subit de différentes déformations par rapport à la forme de l'objet. Le nuage de points capturé est alors décalé dans la direction de la ligne de base entre le laser projecteur et le centre optique (centre focal) du capteur infrarouge [42]. Une procédure de corrélation mesure tous les changements pour tous les points du nuage pour avoir une carte de disparité comme montre la figure 1.13.

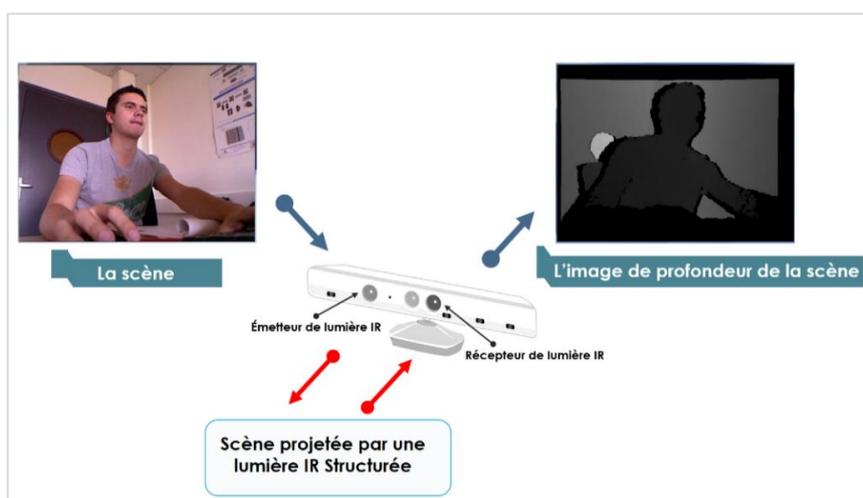


Figure 1.13 : Extraction d'une image de profondeur avec la Kinect

Pour déterminer la coordonnée de la profondeur Z du point k à partir de la disparité, la figure 1.14 illustre la relation entre la distance d'un objet (point k) et de capteur par rapport à un plan de référence.

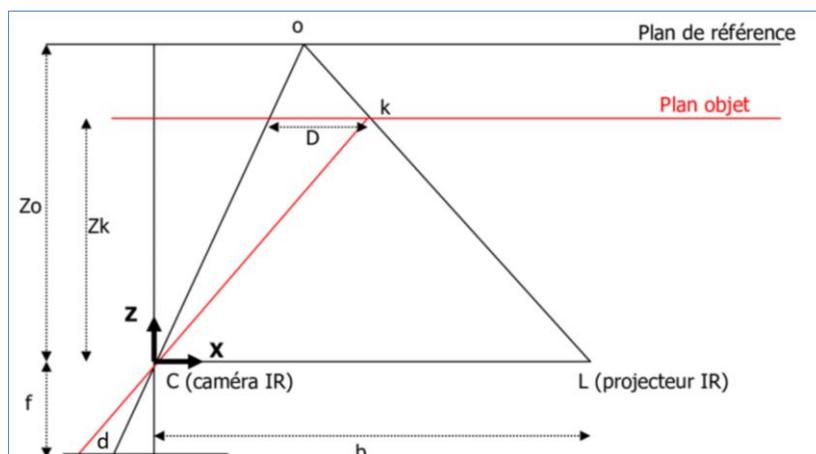


Figure 1.14: Mesure de la valeur de la disparité d [43]. Mise en évidence de la corrélation entre le motif observé et le motif de référence

La distance Z_k , qui correspond à la distance au plan de l'objet visé, se calcule avec l'équation suivante :

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{f}d} \quad (1.9)$$

Z_0, f et b sont des constantes qui peuvent être obtenues à partir de l'étape de calibration. Nous pouvons obtenir les coordonnées X_0 et Y_0 par les équations suivantes :

$$X_k = \frac{-Z_k}{f} (X_k - X_0 + \delta_x) \quad (1.10)$$

$$Y_k = \frac{-Z_k}{f} (Y_k - Y_0 + \delta_y) \quad (1.11)$$

Où X_k et Y_k sont les coordonnées images, X_0 et Y_0 sont les coordonnées du point principal et δ_x et δ_y sont la correction de la distorsion de la lentille.

5.2.2 Extraction du squelette du sujet humain

Pour la reconstruction de la posture humaine sous forme de squelette, la Kinect utilise la méthode détaillée dans [23] des auteurs Shotton *et al.* Leur approche ressemble à celle de la reconnaissance d'objets, en créant à partir des images de profondeur une représentation intermédiaire des parties du corps, qui transforme le problème d'estimation de posture, en un simple problème de classification de pixels, qui permettra de faire une estimation sur la position 3D des articulations du corps (Voir Figure 1.12). Leur travail regroupe les trois étapes de la squelettisation illustré dans la figure 1.15, à savoir: l'extraction du sujet humain, l'estimation de posture, et l'estimation du squelette. Cette approche repose principalement sur deux objectifs de conception clés : l'efficacité et la robustesse des calculs.

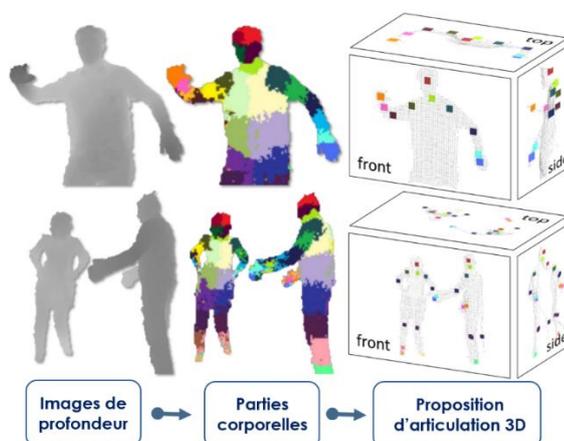


Figure 1.15: Processus de la reconstruction de la posture humaine proposé par Shotton *et al* [23]

La première partie du travail est de pouvoir générer une représentation des différentes parties du corps, à partir des images de profondeur. Pour cela, la méthode utilise un algorithme d'apprentissage pour faire la classification des pixels, en se basant sur la comparaison des caractéristiques de profondeur de deux pixels, dont voici la formule :

$$F_{\theta}(I, x) = d_I \left(x + \frac{u}{d_I(x)} \right) - d_I \left(x + \frac{v}{d_I(x)} \right) \quad (1.12)$$

Tel que :

1. $d_I(x)$: Profondeur du pixel x dans l'image I .
2. $\theta = (u, v)$: Paramètre qui décrit les offsets u et v .
3. $1/d_I(x)$: La normalisation qui assure l'invariance par rapport à la profondeur, vu que dans l'image de profondeur, la taille dépend de la profondeur.

Les caractéristiques d'un pixel seul ne permettent pas de définir la partie du corps à laquelle appartient le pixel, cependant en les intégrant dans une forêt de décision, elles sont suffisantes pour désambiguïser avec précision toutes les parties du corps formées. Shotton *et al* ont utilisé une forêt de décision aléatoire, qui est considéré comme une méthode d'apprentissage supervisée, elle est définie comme un ensemble de N arbres de décision, chacun étant composé de nœuds pères et fils, chaque nœud est constitué d'une caractéristique F_{θ} et d'un seuil τ . Grâce à ces deux dernières, ils font une évaluation du pixel au niveau de chaque nœud afin de déterminer l'un des nœuds fils à suivre jusqu'à arriver au niveau d'une feuille de l'arbre. Cela permet de classer les pixels x dans l'image i . Ils calculent la probabilité d'appartenance du pixel à chacune des classes C et les résultats seront stockés dans un histogramme. Enfin, ils calculent la moyenne des probabilités résultantes de tous les arbres pour avoir la classe à laquelle appartient le pixel :

$$P(c|I, x) = \frac{1}{T} \sum_{t=1}^T P_t(c|I, x) \quad (1.13)$$

En dernier lieu, Shotton *et al* utilisent une technique de recherche de mode, basée sur l'algorithme « Mean-Shift »[66] avec un noyau gaussien pondéré pour calculer l'estimation de la position 3D des articulations. Un estimateur de densité qui représente le nombre de voxels par volume et par partie du corps est utilisé comme mesure, il est défini par :

$$f_c(x) \propto \sum_{i=1}^N w_{ic} \exp \left(- \left| \frac{x - x_i}{b_c} \right|^2 \right) \quad (1.14)$$

Où :

1. x est une coordonnée dans l'espace du monde en 3D.
2. N est le nombre de pixels dans l'image.
3. W_{ic} est une pondération du pixel.
4. x_i est la reprojection du pixel dans l'espace étant donné la profondeur $d_i(x)$.
5. b_c est une largeur apprise pour chaque partie.

Nous remarquons que toutes ces extensions de la technique de base de la méthode Shotton *et al.* mettent en œuvre plus de moyens et plus de calcul pour raffiner le résultat de construction de posture, mais héritent toutes des limitations associées à la mono vision (principalement l'auto occultation). C'est pour cela que nous l'introduisons à partir de la prochaine section l'intérêt de coupler plusieurs capteurs de profondeur.

6 La fusion de données dans les systèmes multi-capteurs

La fusion de données est apparue dans les années 1960 afin de servir des applications militaires. De même, depuis la fin du siècle dernier, la fusion de données a été élargie à des applications robotiques, spatiales, des traitements de signal et d'image.

La fusion des données consiste à déterminer une méthode fiable pour combiner des observations provenant de différents capteurs afin de fournir une description robuste et complète d'un environnement, d'un objet 3D ou d'une silhouette, d'une manière appropriée pour obtenir des informations améliorées, moins coûteuses, de meilleure qualité et plus pertinentes [44].

La fusion de données est un concept que les êtres humains utilisent quotidiennement pour construire une image mentale unifiée de l'environnement et ceci en combinant les perceptions de ses différents sens [44]. Donc nous pouvons dire que la fusion de données multisensorielles est utilisée premièrement par les êtres vivants.

La fusion de données vise l'association, la combinaison, l'intégration et le mélange de multiples sources de données représentant des connaissances et des informations diverses dans le but de fournir une meilleure décision par rapport à l'utilisation séparée des sources de données. Trois éléments sont toujours considérés dans le cadre de la fusion de données : les sources, les algorithmes et le résultat. Le résultat sera composé de données ayant une valeur ajoutée par rapport aux données initiales en provenance de la source [45].

La fusion de données trouve ses applications dans un grand ensemble de domaines tels que la robotique, le contrôle et le commandement militaire, la médecine, la vision robotique, l'interprétation d'images (fusion d'images satellites ou médicales), le contrôle et le monitoring de processus, l'extraction de connaissances dans de grandes banques de données, etc...

En général, la fusion de données présente plusieurs avantages. Ces avantages concernent principalement l'amélioration de l'authenticité ou la disponibilité des données. Dans le premier cas, nous pouvons citer l'amélioration de la détection, la confiance et la fiabilité, ainsi que la réduction de l'ambiguïté des données, tout en étendant la couverture spatiale et temporelle appartiennent à ces dernières catégories de prestations [46].

6.1 Les méthodes de fusion de données

Les méthodes les plus utilisées dans la fusion des données se résument en :

6.1.1 Approche bayésienne

La règle de Bayes est au cœur de la plupart des méthodes de fusion des données. La règle de Bayes permet de déduire un objet ou un environnement d'intérêt décrit par un état A à partir d'une observation B [47].

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (1.15)$$

Les capteurs sont considérés comme un ensemble d'entités capables de fournir une décision à tout instant où chaque capteur donc est vu comme un estimateur bayésien.

Cette approche est relativement simple, mais ne permet pas encore de représenter des modèles complexes évoluant dans le temps [45].

6.1.2 Le filtre de Kalman

Le filtre de Kalman est un filtre bayésien sous l'hypothèse gaussienne et modèles linéaires. C'est un estimateur récursif qui estime les états du système à partir des mesures bruitées tout en minimisant l'erreur quadratique moyenne [48].

Il est utilisé pour le contrôle d'un système dynamique dans la fusion des données multi-capteur. Il est considéré comme une méthode récursive de fusion des données. Il prend aussi en considération les erreurs de modélisation et de mesure en introduisant la notion de probabilité. Le système peut être décrit par un modèle linéaire. Les erreurs associées aux capteurs et au système sont modélisées par un bruit gaussien. Le filtre de Kalman possède un certain nombre

de fonctionnalités qui le rendent parfaitement adapté au traitement des problèmes complexes d'estimation multi-capteurs et de fusion des données. En particulier, la description explicite du processus et des observations permet d'incorporer une grande variété de modèles de capteurs dans l'algorithme de base [49].

6.1.3 Les filtres à particules

Les filtres à particules, aussi connus comme méthodes de Monte-Carlo séquentielles, c'est une méthode qui conduit à une approximation du filtre bayésien en utilisant des méthodes numériques [48]. Ces filtres sont des techniques particulièrement élaborées d'estimation de modèles basées sur la simulation [50]. Ils forment une alternative au filtre de Kalman, cela revient à leur rendu très précis et avec suffisamment d'échantillons.

6.2 Architectures des systèmes de fusion de données multi capteurs

Les systèmes de fusion multi-capteurs doivent avoir une architecture pour contrôler et gérer le processus de fusion. Il existe plusieurs types d'architecture dans la conception de la fusion des données, AL HAGE *et al.* dans [48] ont classifié les systèmes de fusion en :

6.2.1 Architecture centralisée

L'architecture centralisée est plus gourmande en bande passante mais elle profite de l'information globale du système. L'unité centrale doit disposer des informations globales à chaque instant. L'architecture centralisée nécessite une grande bande passante d'un réseau pour envoyer les données brutes des capteurs ainsi qu'une puissance de calcul importante. L'inconvénient du système est d'être non évolutif et peu robuste du fait de la sensibilité aux défauts du processeur central.

6.2.2 Architecture distribuée

Une architecture distribuée peut être considérée comme étant une architecture hiérarchisée à un seul niveau. Les systèmes distribués sont plus modulaires et flexibles contrairement aux architectures centralisées et hiérarchisées qui parfois imposent des coûts de communications et de calculs élevés. Chaque capteur possède un processeur local qui lui permet d'extraire de l'information utile à partir du nœud source. Ceci permet de transmettre moins d'informations et de réduire la charge de calcul à l'unité centrale.

6.2.3 Architecture hiérarchisée

Nous pouvons distinguer deux types d'architecture hiérarchique, les architectures à un seul niveau qui possèdent un seul processeur de fusion et les architectures hiérarchique à plusieurs niveaux dont plusieurs unités de fusion sont utilisées. Pour les deux architectures, les nœuds des niveaux de traitement bas transmettent des informations vers le niveau haut à travers des niveaux intermédiaires. Les informations sont fusionnées et raffinées à chaque niveau jusqu'à l'obtention d'une information globale au plus haut niveau. Le principal avantage de cette architecture est de réduire la charge au niveau de l'unité centrale et améliore la robustesse du système de fusion.

6.2.4 Architecture décentralisée

Un système décentralisé se compose d'un ensemble de capteurs dont chacun possède son propre local et qui, ensemble, ne nécessitent aucune fusion centrale. Par conséquent, chaque nœud fusionne ses données avec l'information communiquée par les nœuds voisins. Ce système est plus souple du fait que les données des capteurs ne se véhiculent pas à un centre de fusion. De même, il est évolutif et robuste aux défauts, de plus la défaillance d'un capteur n'aura pas d'impact sur la structure. Toutefois, l'augmentation du nombre de nœuds entraîne une augmentation du coût de la communication.

7 Conclusion

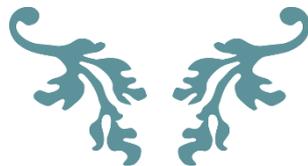
Dans ce chapitre, nous avons passé en revue les notions fondamentales, relatives à notre thème de PFE, telles que le mouvement humain, les types de mouvement humain, la posture et la marche humaine. Nous avons montré l'intérêt de l'analyse du mouvement humain et expliqué son utilité dans le domaine de la capture des mouvements humains, nous avons fait le tour des différentes techniques du capture de mouvement. Trois catégories d'approches se sont dégagées et présentent chacune des avantages et inconvénients. Nous avons par la suite défini le capteur Kinect de Microsoft, son mode de fonctionnement ainsi que le processus de squelettisation d'un sujet humain. Ensuite nous avons présenté les Avatars et leurs utilisations dans le monde virtuel ainsi que les techniques utilisées dans l'animation de ces Avatars à savoir les techniques de cinématique directe et inverse. Enfin nous avons clôturé ce chapitre par la présentation du concept de la fusion de données, des méthodes employées et des architectures des systèmes multi-capteurs.

Dans le chapitre suivant nous allons présenter les systèmes de capture de mouvement basés sur le capteur Kinect de Microsoft ainsi que la proposition de notre système.



Chapitre 2

**Les systèmes multi-Kinect pour
la capture des mouvements**



1 Introduction

Dans ce chapitre, nous présentons une étude et une analyse des systèmes de capture de mouvements humains existants dans la littérature. Cette étude va nous permettre d'une part à connaître leurs performances et d'autre part, à extraire leurs limites de fonctionnement afin d'apporter d'éventuelles améliorations.

2 Domaine d'utilisation de la capture de mouvements

La capture du mouvement humain évolue et progresse selon les possibilités techniques et les développements technologiques existants. Ces dernières années, celle-ci a envahi divers domaines de la science où dans le domaine de la biomécanique et des neurosciences, l'analyse du mouvement humain est utilisée afin de faire progresser les connaissances sur le contrôle sensori-moteur. Ce qui permet de quantifier de manière fiable le mouvement et l'évaluation des résultats d'un traitement donné. La figure 2.1 présente un exemple de capture de mouvement dans le domaine médical.

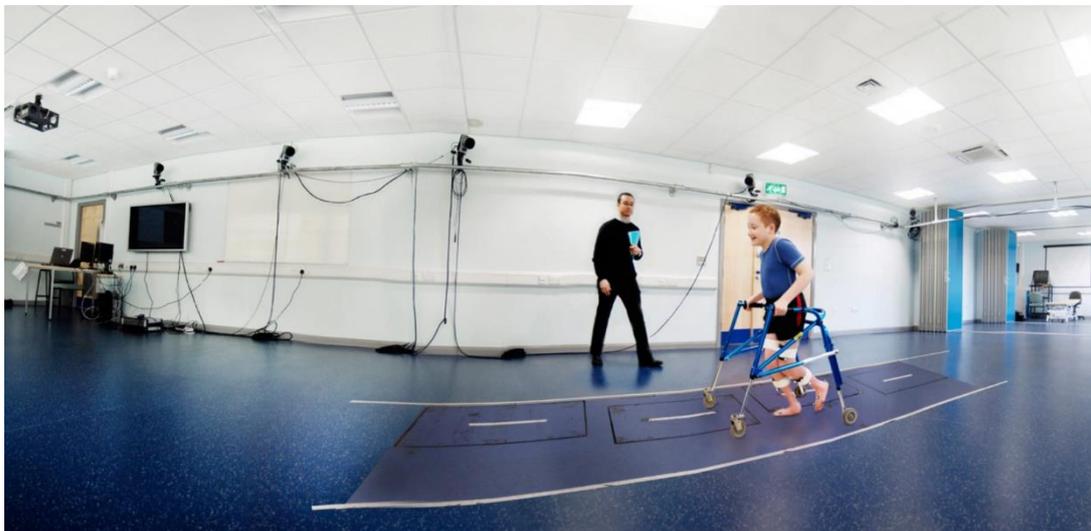


Figure 2.1 : Exemple de capture de mouvement dans le domaine médical [51]

Également, la capture de mouvement s'est largement développée dans le domaine du sport pour améliorer les performances ou prévenir le risque de blessures chez les sportifs dans de nombreuses disciplines comme par exemple : l'Athlétisme, les sports de raquette et de lancer, le cyclisme, la natation, etc. Elle permet d'une part de vérifier que le mouvement est correctement réalisé et qu'il n'est pas traumatisant pour le corps et d'autre part de s'assurer que la blessure n'a pas entraîné des compensations dans le cas d'une reprise après une blessure. Un

exemple de capture de mouvement dans le domaine sportif est donné sur la figure 2.2.



Figure 2.2: Exemple de la capture de mouvement dans le domaine sportif [52]

La danse « l'Art immatériel » qui se concentre sur les aspects esthétiques ou artistiques, les aspects culturels et sociaux, le récit d'histoires (dances symboliques), les significations spirituelles (derviches tourneurs), ... ont aussi utilisé la capture de mouvement comme un moyen pour préserver et transmettre cet art aux générations futures. La figure 2.3 montre un exemple de capture du mouvement dans le domaine de la danse. Dans [53], les auteurs montrent l'existence d'un risque que certains éléments de cette forme de patrimoine culturel immatériel s'éteignent ou disparaissent s'ils ne sont pas sauvegardés et transmis aux générations suivantes. Dans ce contexte, les auteurs ont développé un système de capture, d'analyse et de modélisation des interactions des danses qui pourrait contribuer de manière significative à ce transfert de connaissances.



Figure 2.3 : Exemple de capture de mouvement dans le domaine de la danse [53]

L'utilisation de la capture de mouvement se présente aussi dans le domaine du cinéma, elle est intégrée dans des films réalisés en 3D tel que la saga du Seigneur des Anneaux (2001-2013) avec les personnages de *Gollum* *Smaug* ou encore *King Kong* (2005). Il s'agit d'œuvres dont l'utilisation de la capture de mouvement est particulièrement centrée sur un personnage de l'histoire. La figure 2.4 représente des captures de mouvement utilisées dans le cinéma.



Figure 2.4 : Exemple de capture de mouvement dans le domaine du cinéma [54]

Le domaine des jeux vidéo a également utilisé cette technique pour plus d'interaction des joueurs sous diverses formes intellectuelles ou physiques comme le représente la figure 2.5 par un exemple de capture de mouvement dans le domaine des jeux vidéo [55].

La capture du mouvement permet aux personnages d'être très réalistes et d'avoir certaines expressions faciales et gestuelles de véritables acteurs humains. Citons par exemple les jeux de console Xbox 360 qui utilisent le capteur Kinect pour la capture du mouvement des joueurs et l'interaction avec le monde virtuel.



Figure 2.5: Exemple de capture de mouvement dans le domaine des jeux vidéo [55]

3 L'étude et l'analyse des systèmes similaires existants

La capture de mouvement sans marqueurs est généralement utilisée pour les études dans le domaine de la vision par ordinateur. La plupart de ces études sont portées sur l'utilisation de plusieurs capteurs pour acquérir les mouvements du squelette ou l'apparence de l'utilisateur humain. Certains travaux adoptent l'analyse des formes et des couleurs sur une séquence d'images multi-vues pour en déduire le mouvement du squelette. Cependant, cette approche nécessite souvent un large ensemble d'images d'entraînement capturées auprès de l'utilisateur. D'autres utilisent un modèle humain 3D très détaillé pour la capture des mouvements humains, mais en général les données préalables sont difficiles à préparer à l'avance pour un utilisateur.

Récemment, l'émergence des capteurs de profondeur peu coûteux, tels que Microsoft Kinect, a permis de capturer le mouvement humain en temps réel. Au même temps, un système utilisant un seul capteur Kinect souffre souvent de problèmes d'occlusion des articulations et d'orientation du corps. Lorsque des parties du corps ne sont pas observables par le capteur (tel le cas où l'utilisateur se tourne sur le côté), les données de profondeur ne peuvent pas être récupérées, ce qui conduit à un échec d'estimation de la posture. Pour cette raison, l'utilisateur doit être face au capteur de telle sorte que son corps soit visible.

Pour résoudre le problème d'occultation de certaines parties du corps, plusieurs études ont adopté l'utilisation de plusieurs capteurs Kinect. Berger *et al.* dans [56] ont proposé une méthode pour utiliser quatre capteurs Kinect afin d'extraire des silhouettes et de capturer les mouvements. Zhang *et al.* dans [57] ont appliqué des techniques de filtrage des particules et d'échantillonnage de partitions pour suivre les postures. Cependant, ces méthodes s'appuient sur des données non squelettiques et un processus d'optimisation qui dépend de la silhouette ou du modèle pour estimer les postures.

En outre, Williamson *et al.* dans [58] ont proposé un système d'entraînement des soldats utilisant plusieurs capteurs Kinect qui peuvent capturer les mouvements, même lorsque l'utilisateur tourne à 360°. Kaenchan *et al.* dans [59] ont analysé les mouvements de la marche en se basant sur les positions moyennes des articulations suivies. Néanmoins ces systèmes souffrent de l'incertitude des données résultantes de la moyenne des différents squelettes à cause des données erronées fournis par le capteur.

Kitsikidis *et al.* dans [60] ont utilisé trois capteurs Kinect pour récupérer les mouvements de danse lente par un découpage du squelette en cinq parties différentes, et de générer

automatiquement un vocabulaire de posture (*codebook*) pour chaque partie. Ils appliquent l'algorithme HCRF (*Hidden State Conditional Random Field*) pour la reconnaissance des postures de danse. Ce système est limité sur le mouvement lent et ne supporte pas les mouvements dynamiques rapides.

Moon *et al.* dans [61] ont utilisé la méthode basée sur le filtrage de Kalman pour modifier et mélanger des données Kinect précises. En revanche, ils n'ont pas réussi à capturer les événements de mouvement à 360° et les mouvements résultants étaient trop simples. Jo *et al.* dans [62] ont proposé un système utilisant plusieurs capteurs Kinect pour suivre plusieurs utilisateurs. Cependant, ils se sont concentrés sur le suivi des positions des utilisateurs multiples au lieu de récupérer leurs mouvements.

Les auteurs de A.Naveed dans [63] ont utilisé quatre capteurs Kinect pour capturer les mouvements des boxeurs et la marche d'un utilisateur humain. Le système suit le visage de l'utilisateur pour déterminer le capteur central. Les données des articulations des autres capteurs sont utilisées pour récupérer les articulations du squelette que le capteur central ne parvient pas à suivre.

De même, Kim *et al.* dans [64] ont utilisé la méthode ICP (*Iterative Closest Point*) pour constituer une représentation unique du squelette à partir des systèmes de coordonnées des nuages de points fournis par les différentes Kinect. Cependant le système n'est pas fiable pour les mouvements rapides des utilisateurs.

Afin de positionner notre approche par rapport à chaque travail de la littérature, une étude comparative non exhaustive est présentée dans le Tableau 2.1.

Tableau 2.1: Tableau comparative des travaux sur la capture du mouvement

	Berger et .al [56]	Zhang et .al [57]	Williamson et .al [58]	Kaenchan et .al [59]	Kitsikidis et .al [60]	Moon et .al [61]	Jo et .al [62]	A.Naveed [63]	Kim et .al [64]
Nombre de Kinect	04	02	04	03	03	05	04	04	[04-08]
Cameras	RGB -D	D	D	D	D	D	RGB -D	RGB -D	D
Multi Vue	Non	Non	Non	Oui	Oui	Oui	Oui	Oui	Non
Contenu des séquences	MOCAP	MOCAP	Entrainement des Soldats	Marche Humaine	Analyse de la dance	MOCAP	MOCAP	la boxe marche	Dance et Taekwondo
Calibration	Oui	Oui	Non	Oui	Oui	Oui	Non	Oui	Oui
Fusion de données	Non	Oui	Oui	Oui	Oui	Oui	Non	Oui	Oui
Face détecte	Non	Non	Non	Non	Non	Non	Non	Oui	Non
Angle de capture	180°	360°	360°	360°	360°	180°	360°	360°	360°
Fixation Kinect	Oui	Non	Oui	Non	Non	Non	Oui	Oui	Non

*D : Camera de profondeur *RGB : Camera RGB

D'après le tableau 2.1, nous notons que tous les systèmes mentionnés précédemment s'accordent sur l'utilisation de multi-Kinect pour la capture du mouvement humain dans les différents domaines afin de limiter les effets résultants du problème de l'auto-occlusion. Le tableau conclue que :

- La précision et la fiabilité de l'estimation de la posture humaine dépend du nombre de capteurs utilisés, de leurs emplacements, de leurs calibrations ainsi que de la manière de fusion des données acquises par les différents capteurs.
- La camera RGB permet la vérification de l'exactitude des données de la caméra de profondeur.
- La visualisation multiple joue un rôle très important. La plupart des systèmes cités dans le tableau fournissent plusieurs vues afin d'offrir une bonne perception, pour permettre l'analyse du mouvement humain sous différents angles.
- Le calibrage des caméras est essentiel dans la plupart des systèmes de capture du mouvement, car cette opération établit le lien entre le repère du monde réel et le repère du capteur. Plusieurs techniques sont bien établies pour réaliser cette opération. Il existe toutefois certaines situations pour lesquelles les techniques usuelles ne sont pas assez précises.
- La phase de la fusion représente le processus principal de la capture du mouvement. Elle consiste à combiner les données provenant de différentes sources afin de limiter l'erreur produite par les capteurs Kinect lorsque le corps humain n'est que partiellement observable ou que la fiabilité de l'information n'est pas garantie.
- La technique de la détection de visage est vraiment nécessaire car l'algorithme d'extraction du squelette ne fait pas de différence entre une personne vue de face et une personne vue de dos, c'est-à-dire que les membres supérieurs/inférieurs de gauche et les membres de droite seront inversés si la personne est vue de dos.

D'autre part, Ousmer dans [65] a proposé un système de capture des mouvements humains pour l'animation d'un Avatar en utilisant un seul capteur Kinect. Il est constitué de deux principaux modules comme montre la figure 2.6.

Le premier module de reconnaissance et de suivi a pour but d'extraire la posture humaine sous forme de données squelettiques.

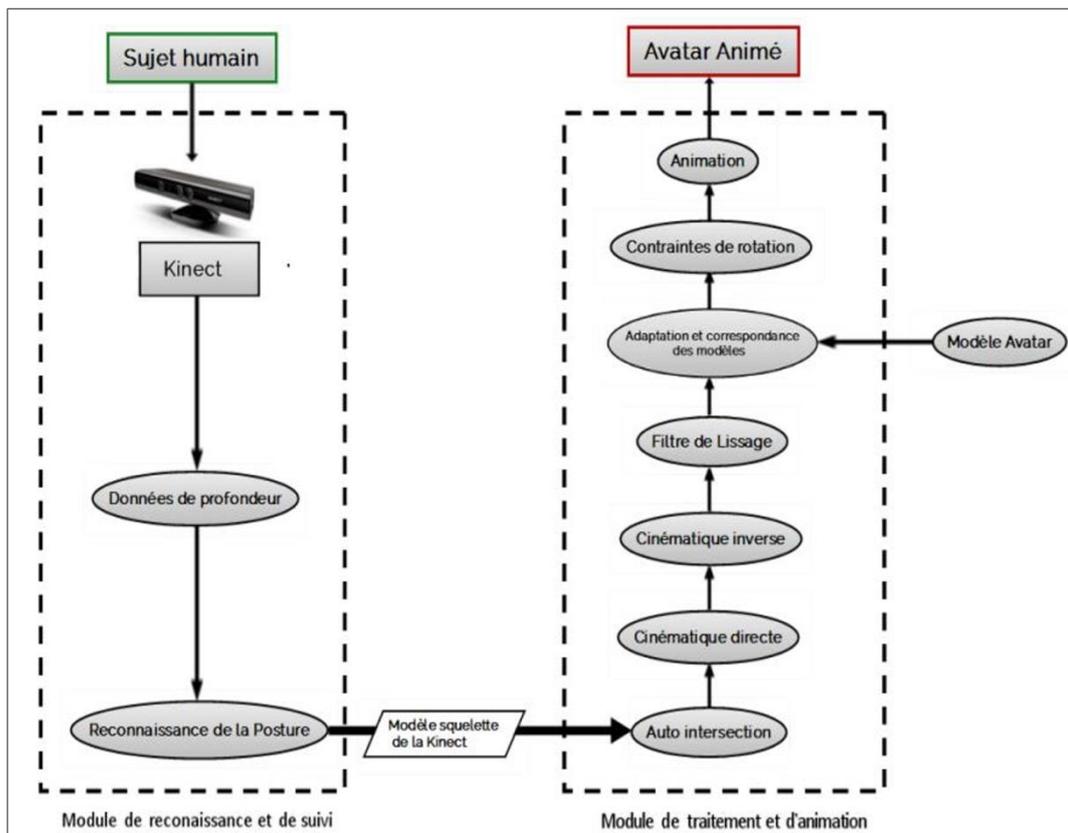


Figure 2.6 : Schéma du système de capture de mouvements pour l'animation d'Avatars [65]

Dans le deuxième module, il effectue, avant l'animation de l'Avatar, une série de traitements sur les données squelettiques brutes afin de corriger et d'atténuer les bruits qu'elles contiennent, à savoir :

- L'auto-intersection qui permet de résoudre les problèmes d'intersection entre les membres extérieurs du squelette (mains, poignets) et le buste.
- La cinématique directe qui s'effectue pour corriger les positions et les rotations des articulations des extrémités du squelette (mains, pieds, tête) si ces derniers ont des valeurs imprécises ou inexacts.
- La cinématique inverse qui est utilisée pour corriger les positions des différentes parties internes du squelette à partir des extrémités des membres lorsqu'elles ne sont pas bien détectées par le capteur Kinect.
- Le filtre de lissage qui sert à assurer la fluidité et la continuité des mouvements lors de l'animation de l'Avatar dans le temps entre les frames.

Tout de même, ayant utilisé un seul capteur pour la capture des mouvements humains, ce système souffre aussi des limites suivantes :

- Un manque de précision et de robustesse avec une instabilité liée à la latence.
- Une restriction de la liberté du mouvement (rotation) du sujet humain dans la scène à cause du problème d'occultation.
- Lorsque des parties du corps ne sont pas observables par le capteur, les données de profondeur ne peuvent pas être récupérées, ce qui conduit à un échec d'estimation de la posture.
- Ce système oblige l'utilisateur de se tenir face au capteur tout au long du suivi pour pouvoir observer la totalité du corps, d'où une réduction et une limite de flexibilité du système.

4 L'approche proposée

L'étude de l'existant que nous avons réalisée, nous a permis de comprendre les approches utilisées dans les différents systèmes existants dans le domaine de la capture du mouvement humain, d'en extraire leurs limites et leurs lacunes, de mieux collecter leurs paramètres et de réfléchir aux améliorations possibles et nécessaires de ces systèmes dans le contexte abordé dans ce projet de fin d'études.

L'objectif de notre travail étant la conception d'un système d'analyse du mouvement à faible coût qui sera utilisé en rééducation fonctionnelle (analyse de la marche par exemple). En premier lieu et d'après l'étude des systèmes existants dans la littérature, l'utilisation d'un ensemble de capteurs Kinect est plus que nécessaire (vision en 360°). Chaque capteur Kinect est lié à un PC local « client » interconnecté au réseau local permettant l'échange et le transfert des données acquises par des capteurs d'une manière fiable et rapide à l'unité centrale « Serveur » afin de suivre en temps réel les mouvements rapides et dynamiques d'un sujet humain et de fusionner les données squelettiques limitant ainsi les problèmes d'occultations. En second lieu, les données brutes acquises par les capteurs ne sont pas très précises, d'où la nécessité d'un ensemble de traitements (l'auto-intersection, la cinématique directe et le filtre de lissage) qui se réalisent dans chaque PC local « client » avant d'être transmises à l'unité centrale du système « serveur » pour augmenter leur précision et leur fiabilité et les rendre plus réalistes. Une étape importante avant la fusion des données traitées consiste à calibrer les capteurs pour assurer leurs homogénéités lors de la fusion. Ensuite, l'élaboration d'une méthode fiable et robuste de fusion des données traitées toute en garantissant la fluidité des traitements pour la préservation de l'aspect temps réel de la capture des mouvements. Enfin, l'animation différée des mouvements réels capturés dans un monde virtuel par le biais d'un Avatar.

Notre approche permet d'une part au sujet humain de se déplacer librement dans la scène sur un panorama de 360°, sans être figé devant un seul capteur Kinect et d'autre part d'offrir une vision multi-façades du sujet humain. Le système que nous proposons est basé sur plusieurs processus : l'acquisition et le prétraitement des données squelettiques, le calibrage, la transformation des données traitées, la reconstruction de la posture par la fusion des données squelettiques, l'animation (« Temps réel » « Différée ») et la sauvegarde de cette dernière pour une analyse future.

5 Conclusion

Dans ce chapitre, nous avons présenté un aperçu sur les domaines d'utilisation de la capture des mouvements humains. Nous avons par la suite, passé en revue les systèmes existants dans la littérature où nous avons listé leurs lacunes et limites. Ensuite, nous avons spécifié l'approche que nous avons adopté pour la réalisation de notre système de capture de mouvements à faible coût qui sera utilisé pour un premier temps en rééducation fonctionnelle.

Dans le chapitre suivant, nous présenterons l'architecture globale du système et que nous proposons et nous détaillerons la conception des différents modules qui la constitue.



Partie II

**Système Multi-Kinect
développé pour la capture de
mouvements**





Chapitre 3

Conception



1 Introduction

Ce chapitre est consacré à la présentation de la conception de notre système d'animation d'Avatars par l'analyse de mouvement d'un être humain en temps réel. Nous présentons d'abord une vue globale de l'architecture proposée qui fera l'objet d'une description détaillée, ainsi que l'ensemble des principaux modules qui la composent, passant par la modélisation UML via ses différents diagrammes. Nous terminons ce chapitre en détaillant les méthodes que nous avons utilisées pour l'acquisition et le prétraitement des données, ainsi que le calibrage, la fusion des données squelettiques et l'animation d'un Avatar dans un environnement virtuel.

2 Architecture du système proposé : Présentation globale de la solution

La solution proposée comprend plusieurs modules : d'acquisition, de traitement et d'affichage d'information et de données. Nous avons vu, à travers l'étude menée au chapitre précédent, que l'utilisation simultanée de plusieurs capteurs Kinect ajoute des complications pour atteindre notre objectif. Ceci, nous a amené à utiliser une approche modulaire dans la conception du système proposé, afin de contourner certaines limitations liées au logiciel et au matériel utilisés. La première limitation est liée à l'utilisation du système d'exploitation Windows qui permet de piloter qu'un seul capteur Kinect à la fois. La deuxième concerne le matériel, où notre choix est porté sur l'utilisation de machines de type PC, les performances de ces dernières ne peuvent pas répondre aux exigences des traitements, temps réel, requis par notre application pour une solution avec plusieurs capteurs Kinect. Pour ce faire notre système est alors basé sur l'utilisation de l'approche où chaque capteur Kinect est piloté par une unité de traitement à part.

Notre système sera donc composé de deux sous-systèmes :

- Le premier concerne la partie client qui sert à l'acquisition, les prétraitements et le transfert des données squelettiques récupérées par le capteur.
- Le deuxième consiste en partie serveur qui sert à recevoir les données fournies par le premier sous système, à utiliser les paramètres issus du calibrage des capteurs, à fusionner les données pour ensuite faire animer un Avatar dans un environnement virtuel.

Pour assurer la communication et l'échange des données, en temps réel, entre les deux sous-systèmes, nous utilisons l'architecture 2-tiers Client/serveur où chaque capteur Kinect est relié à une unité de traitement à part (client) qui communique avec le serveur. Cette architecture

désigne un mode de communication organisé par l'intermédiaire d'un réseau entre différents modules du système que nous proposons.

Un des paramètres pris en considération lors de la conception de notre architecture est : celle-ci doit être évolutive, où il est très facile de rajouter ou d'enlever des clients. Cela nous permet de gérer le nombre des capteurs utilisés ainsi que la manipulation du champ visuel de l'ensemble des capteurs pour une meilleure animation. Ça permet aussi de centraliser les données dans un seul endroit (serveur) pour effectuer les traitements sur les données squelettiques reçues par les différents clients. Un autre point essentiel est le traitement en temps réel, c'est-à-dire, le suivi en temps réel de la pose du sujet humain durant l'expérimentation. Les images de profondeur et de couleur RGB captées par une Kinect sont transmises vers l'ordinateur (client) via le port USB, traitées selon les besoins ensuite les données résultantes seront envoyées à l'ordinateur principal (serveur) par des interfaces de connexion sockets qui communiquent via le protocole TCP/IP. Notre compartimentation en sous-systèmes est aussi mise à profit par la redistribution des traitements ce qui nous assure une meilleure latence, une fluidité et une stabilité de notre système.

Le schéma global de notre système est illustré dans la figure 3.1, ainsi que son déroulement depuis l'acquisition des données squelettiques jusqu'à l'animation finale de l'Avatar dans un monde virtuel après analyse de mouvement d'un sujet humain dans le monde réel. Il peut être divisé en trois principaux modules :

- 1- Un module d'acquisition, de suivi et de prétraitement de la posture humaine ainsi que la détection de son visage en temps réel.
- 2- Un module de calibrage afin d'obtenir une bonne correspondance entre les capteurs.
- 3- Un module basé sur la fusion des données capturées pour estimer les parties du corps occultées du sujet humain pour ensuite animer un Avatar dans le monde virtuel.

Comme le présente la figure 3.1, notre système détecte le sujet humain dans le monde réel, puis effectue un ensemble de procédures de traitements arrangées en plusieurs étapes pour obtenir les informations réelles sur le mouvement du sujet humain pour bien le dupliquer sur un Avatar.

Le module d'acquisition, du suivi et des prétraitements de la posture humaine est réalisé par la partie client. Chaque capteur Kinect est rattaché à un seul client, ce qui constitue le réseau de capteurs Kinect, Il permet dans un premier temps, de fournir les données squelettiques du sujet humain (utilisateur), suivant sa position par rapport à ce dernier, qui seront ensuite corrigées et filtrées localement des bruits et des erreurs qu'elles comportent. Et dans un second temps, le client permet aussi de fournir les données de profondeur et de couleur RGB pour

l'étape de vérification de l'état de détection du visage de l'utilisateur, ce qui nous permettra de savoir l'orientation de son corps par rapport à l'ensemble des capteurs utilisés.

Tandis que les modules de calibrage et de l'animation par la fusion de données, sont réalisés par le serveur. Uniquement, le serveur accueille les données squelettiques envoyées par ses clients et les fusionnent pour reconstruire la posture 3D complète de l'utilisateur pour ensuite la faire correspondre à un Avatar dans un monde virtuel. L'Avatar suivra les mêmes mouvements que le sujet humain exerce dans le monde réel. L'orientation du sujet humain durant l'expérimentation est mise à jour en fonction de l'orientation de son visage détecté par les clients du réseau. Les états de détection du visage de chaque client sont envoyés au serveur à chaque frame.

Notre système utilise des matrices de transformations entre les différents capteurs. Ces matrices sont calculées lors de l'initialisation du système. Elles sont utilisées pour transformer les données squelettiques d'un repère (capteur) à un autre.

En dernier lieu, le serveur fusionne les données résultantes de l'étape précédente pour ensuite adapter le squelette fusionné au squelette de l'Avatar pour avoir une animation plus réaliste et précise copie de celle de sujet humain.

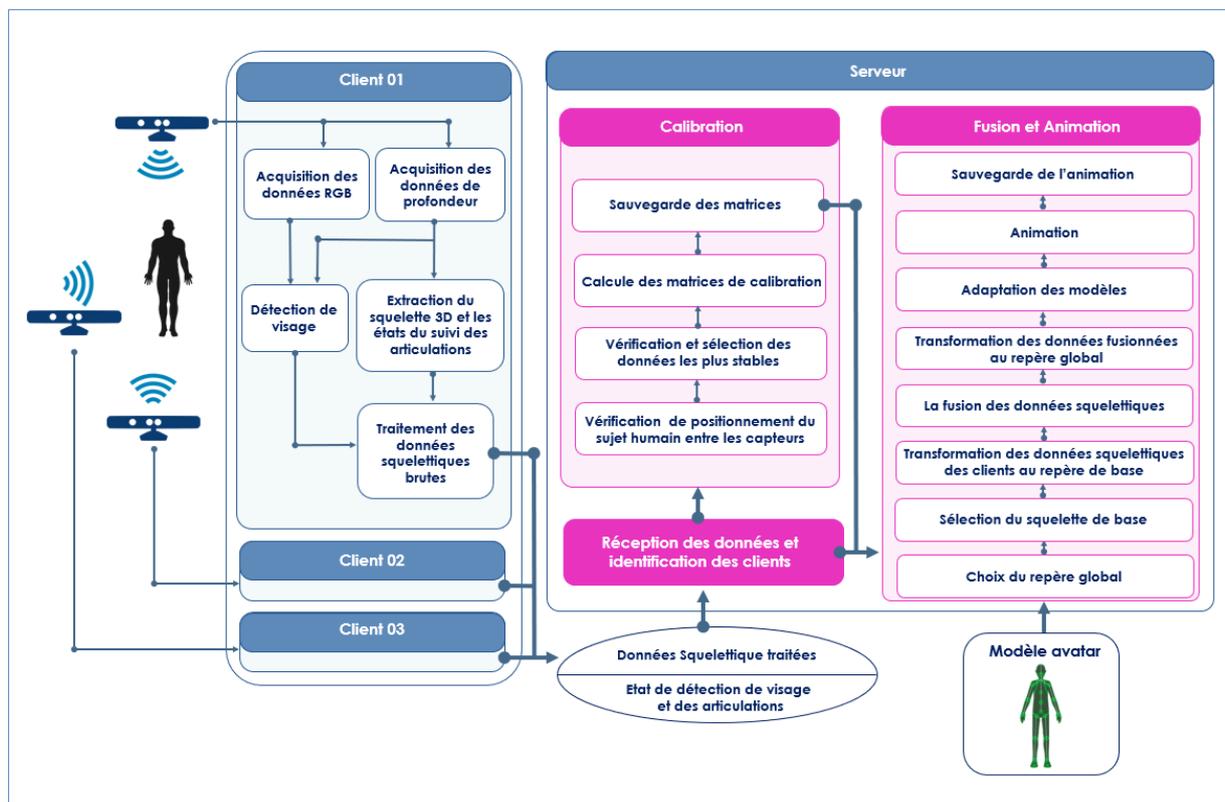


Figure 3.1 : Architecture globale du système proposé.

2.1 Acquisition et prétraitements des données

Le module d'acquisition des données, du suivi et des prétraitements de la posture humaine permet, en premier lieu, d'obtenir, à partir de chaque capteur Kinect du réseau, les informations de couleurs et de profondeur du sujet humain. Chaque client applique simultanément et en parallèle le processus de *Shotton et al.* sur les données de profondeur, pour extraire les données squelettiques du sujet humain et d'estimer les positions 3D de ses articulations dans l'espace tridimensionnel.

En second lieu, nous utilisons la méthode de détection de visage de Nikolai *et al.* [67] pour la reconnaissance de l'orientation du corps du sujet humain par rapport aux capteurs du réseau. Ce choix vient pour solutionner les problèmes des erreurs qui augmentent quand le sujet humain s'éloigne de l'axe de vision du capteur. Cette méthode se base sur les données de couleur et de profondeur, elle est accessible via le SDK Kinect de Microsoft.

Par la suite un ensemble de traitements, détaillé dans la section (§5.1), est appliqué localement (Client) sur chaque squelette avant d'être transmis au serveur. Le premier traitement est l'auto-intersection qui aide à la correction des intersections entre les articulations des deux mains et du buste. Ensuite le traitement de la cinématique directe est appliqué pour éliminer le problème des valeurs imprécises des positions des extrémités des membres supérieurs et inférieurs, lorsque celles-ci sont inférées (non reconnues) et/ou non détectées par la Kinect. Ce processus de traitement de données se termine avec le filtre de lissage des données pour avoir une fluidité et une transition homogène dans le mouvement final de l'Avatar.

2.2 Calibrage

Le calibrage des caméras est l'une des pierres angulaires de notre système. Il permet de trouver les correspondances et le passage entre les repères des capteurs du réseau.

Puisque le sujet humain est appelé à se mouvoir dans un espace réel 3D, la connaissance continue dans le temps de la position exacte des points particuliers de son corps est très importante. Pour obtenir cette connaissance, l'utilisation du calibrage est nécessaire. Il délivra les différentes matrices de passage entre repères capteurs de ces points particuliers vus par ces derniers.

Nous avons appliqué un calibrage fort sur les points des jointures du squelette acquis par les clients (capteur Kinect). Afin d'avoir les relations sous forme de matrices de transformation entre toutes les combinaisons des repères capteurs (Kinect) et de pouvoir faire des estimations des positions 3D des articulations d'un repère vers les autres repères comme le montre la figure 3.2.

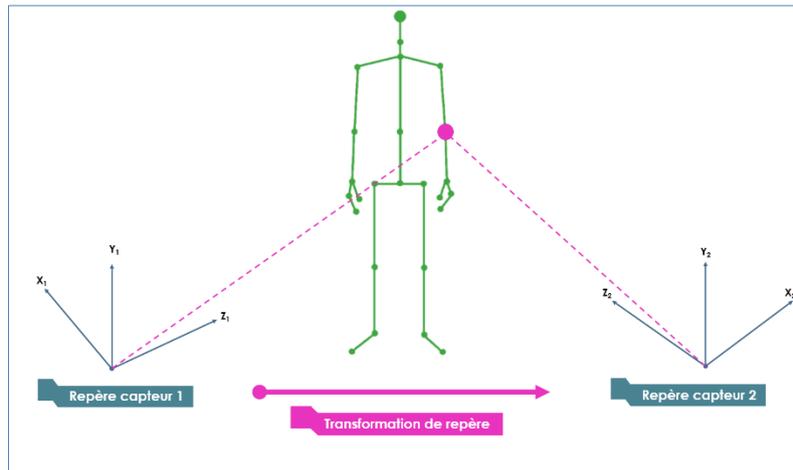


Figure 3.2 : Relation entre les points de vue de deux repères capteurs d'une seule articulation d'un même squelette

2.3 Ajustement et fusion des données pour l'animation d'Avatar

Avant de faire mouvoir notre Avatar dans la scène virtuelle par les mouvements réels de l'utilisateur humain, nous avons suivi plusieurs phases de traitements comme le montre la figure 3.3. Premièrement, nous avons utilisé les données récupérées, via le réseau, par les clients pour choisir lequel parmi eux celui qui sera considéré comme le repère de base. Les positions 3D des joints/articulations non détectés seront estimées à partir des autres clients à l'aide des matrices de transformations entre repères des capteurs calculées dans le module de calibrage.

Ensuite, nous avons appliqué l'approche de fusion de données en combinant les données reçues par les différents capteurs. Ces données peuvent être plus ou moins fiables. Cette approche est souvent nécessaire lorsque le sujet humain n'est que partiellement observable ou que la fiabilité de l'information n'est pas garantie.

Afin de réduire l'incertitude sur les mesures de position des articulations du squelette du sujet humain, nous fusionnons (choisir la donnée la plus juste) les données estimées avec les données du repère de base pour assurer la fiabilité et la couverture totale des positions des articulations. En effet, ce processus nous permettra d'obtenir un squelette reconstruit complet composé uniquement d'articulations bien suivies avec des mesures sûres.

L'algorithme d'extraction du squelette développé par Microsoft fournit un très bon résultat en temps réel, mais il n'est pas suffisant pour une application comme la nôtre. Le fait que la méthode de *Shotton et al.* qui est basée sur des images de profondeur acquises à partir d'un seul point de vue (un capteur) est trop limitatif pour notre application, notamment à cause des occlusions.

Notre approche est conçue dans le but d'augmenter l'angle de vue et réduire l'impact des occlusions par la redondance de l'information en utilisant ce même algorithme selon plusieurs points de vue. Nous prenons l'avantage de cela pour augmenter la couverture (espace observé), la fiabilité (identification) et la précision (mesure de position) du système de suivi de mouvement. Avec ces améliorations, le choix d'utiliser l'algorithme d'estimation de pose du squelette proposé par Microsoft devient une solution efficace et adaptée à nos besoins.

Par la suite, à chaque frame les données des squelettes fusionnés et corrigés du sujet humain, seront affectées aux squelettes de l'Avatar présélectionné par l'opérateur. Cela nous permet d'avoir une animation réaliste, naturelle et fluide de cet Avatar dans le monde virtuel.

2.4 Sauvegarde et représentation graphique de données squelettiques

Durant la phase d'animation, notre système fournit à l'opérateur la possibilité de sauvegarder l'animation des mouvements exercés par le sujet humain. Nous l'avons mis en œuvre par la collection des données de position des articulations corrigées et fusionnées par le module d'ajustement et fusion des données pour l'animation d'Avatar.

Pour atteindre cet objectif nous avons élaboré une structure de données simple et efficace permettant de sauvegarder les animations des mouvements exercées par sujet humain qui seront utiles pour une utilisation ultérieure à savoir la consultation des séances d'examens de rééducation des patients. Une autre fonctionnalité importante fournie par le module de sauvegarde est l'enregistrement des notes et des remarques, du personnel médical, concernant l'évolution de la santé du sujet humain traité.

Les données sauvegardées seront aussi utilisées pour la représentation graphique 3D des articulations qui permet au médecin de visualiser séparément les différentes parties du corps par la sélection des articulations voulant les animer.

2.5 Animation différée

Le présent module repose sur les données obtenues du module de sauvegarde. À la demande du médecin/opérateur, ces données sont réutilisées pour animer un Avatar afin de lui permettre de revoir les mouvements effectués par le sujet humain durant ses séances antérieures de rééducation. En effet ce module offre un suivi de l'évolution du mouvement du sujet humain (le patient dans ce cas) en fonction du temps permettant au médecin de le comparer par rapport aux différentes séances déjà effectuées et de déterminer les anomalies et les handicaps dans les mouvements (la marche par exemple) du patient.

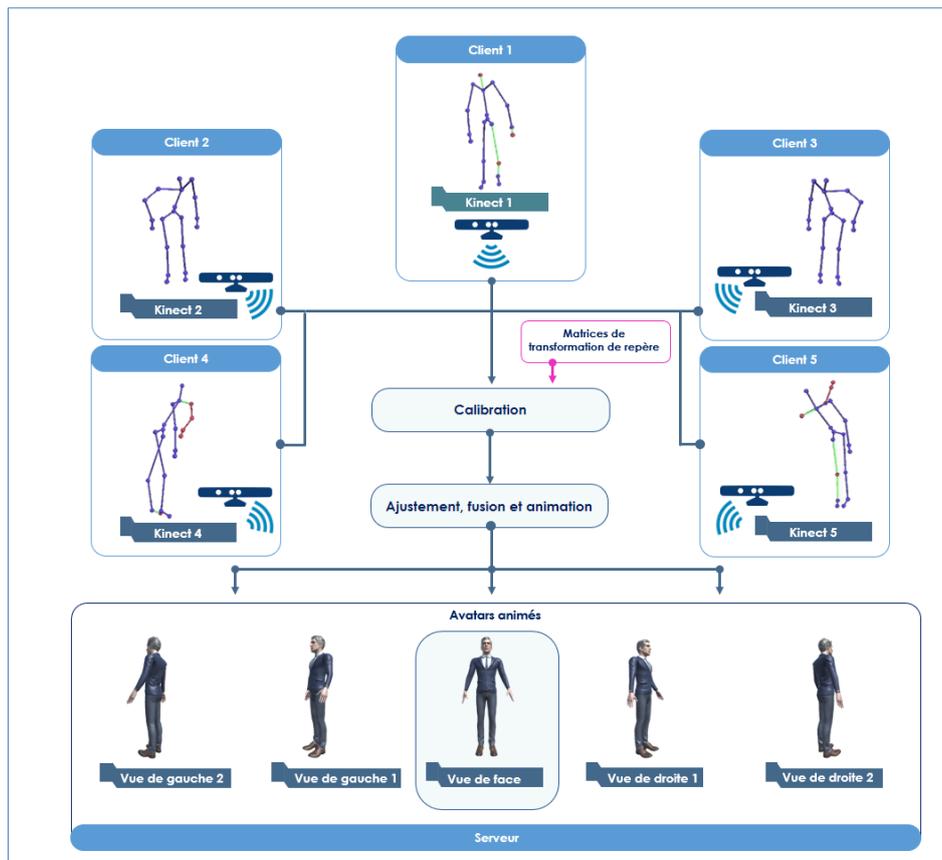


Figure 3.3 : Processus global de l'animation par la fusion de données

3 Modélisation UML

Lors de cette partie, nous allons identifier les principaux diagrammes utilisés pour mettre en œuvre l'architecture proposée. L'objectif fondamental de la modélisation est de faciliter la conception et le développement du système. Elle permet d'organiser les étapes et les différents modules en facilitant et en réduisant leurs complexités. Plusieurs démarches de modélisation existent dans la littérature. Nous adoptons dans notre travail une approche objet basée sur un outil de modélisation UML.

En fait, UML (*Unified Modeling Language*) est un standard ouvert contrôlé par l'OMG. UML est issu de l'unification de nombreux langages de modélisation graphique orientée objet. Il unifie à la fois les notations et les concepts orientés objets.

Pour la modélisation de notre système, nous avons opté pour l'utilisation de certains diagrammes, chacun permettant de visualiser le système proposé avec une vue particulière : diagramme de cas d'utilisation, diagramme de classes, diagramme de séquence et diagramme d'activité.

3.1 Diagrammes de cas d'utilisation

Les spécifications fonctionnelles nous permettent d'extraire un ensemble de cas d'utilisations qui peuvent être exécutés à travers l'utilisation du système.

Le diagramme de cas d'utilisation nous permet d'illustrer les interactions entre le système et ses différents acteurs. Nous avons opté pour ce diagramme car chaque cas d'utilisation correspond à une fonction métier du système, selon le point de vue d'un de ses acteurs. L'acteur principale qui interagit avec notre système c'est le médecin qui est l'utilisateur final du produit. Il utilise le système pour le suivi de ses patients et leurs examens effectués.

Comme notre système est subdivisé en deux sous-systèmes, nous pouvons distinguer d'une part un ensemble d'interactions entre le médecin et la machine dite serveur qui représente le sous-système d'animation par fusion et d'une autre part entre le médecin et les machines dite clientes qui représente le sous-système d'acquisition et prétraitements.

D'après le diagramme de cas d'utilisation représenté dans la figure 3.4, nous pouvons identifier divers cas d'utilisation effectués par médecin/opérateur avec le sous-système d'acquisition et prétraitements, à savoir :

Pour démarrer le système, le médecin/opérateur relie le capteur Kinect à la machine cliente

- **Connexion au serveur** : chaque machine client peut se connecter au serveur via le protocole TCP/IP en spécifiant son adresse comme elle peut se déconnecter à la fin de son utilisation pour être détachée du système.
- **Acquisition des données** : Afin d'acquérir les données squelettiques et l'état de détection du visage, le capteur Kinect connecté à la machine client extrait ses données en utilisant les images de profondeurs et RGB captées du monde réel.
- **Traitement des données squelettiques** : Après l'étape d'acquisition, des filtres de

correction sont appliqués sur les données squelettiques comme l’auto-intersection, la cinématique directe et le filtre de lissage. Ces traitements servent à l’élimination du bruit et d’augmenter la précision et la fiabilité des données.

- **Communication avec le serveur** : les principales données envoyées au serveur sont l’état des capteurs, l’état de suivi du sujet humain, les données squelettiques traitées et l’état de détection de vissage.
- **Mise à jour de l’état du capteur Kinect** : La machine client assure la mise à jour de l’état de suivi du sujet humain et l’état du capteur (connecté, non connecté).
- **L’animation d’Avatar** : Après avoir recueilli les données squelettiques fournies par le capteur Kinect, un ensemble de traitement est appliqué à ces dernières. Les données traitées seront adaptées au squelette de l’Avatar afin d’avoir une animation réaliste dans la scène virtuelle.

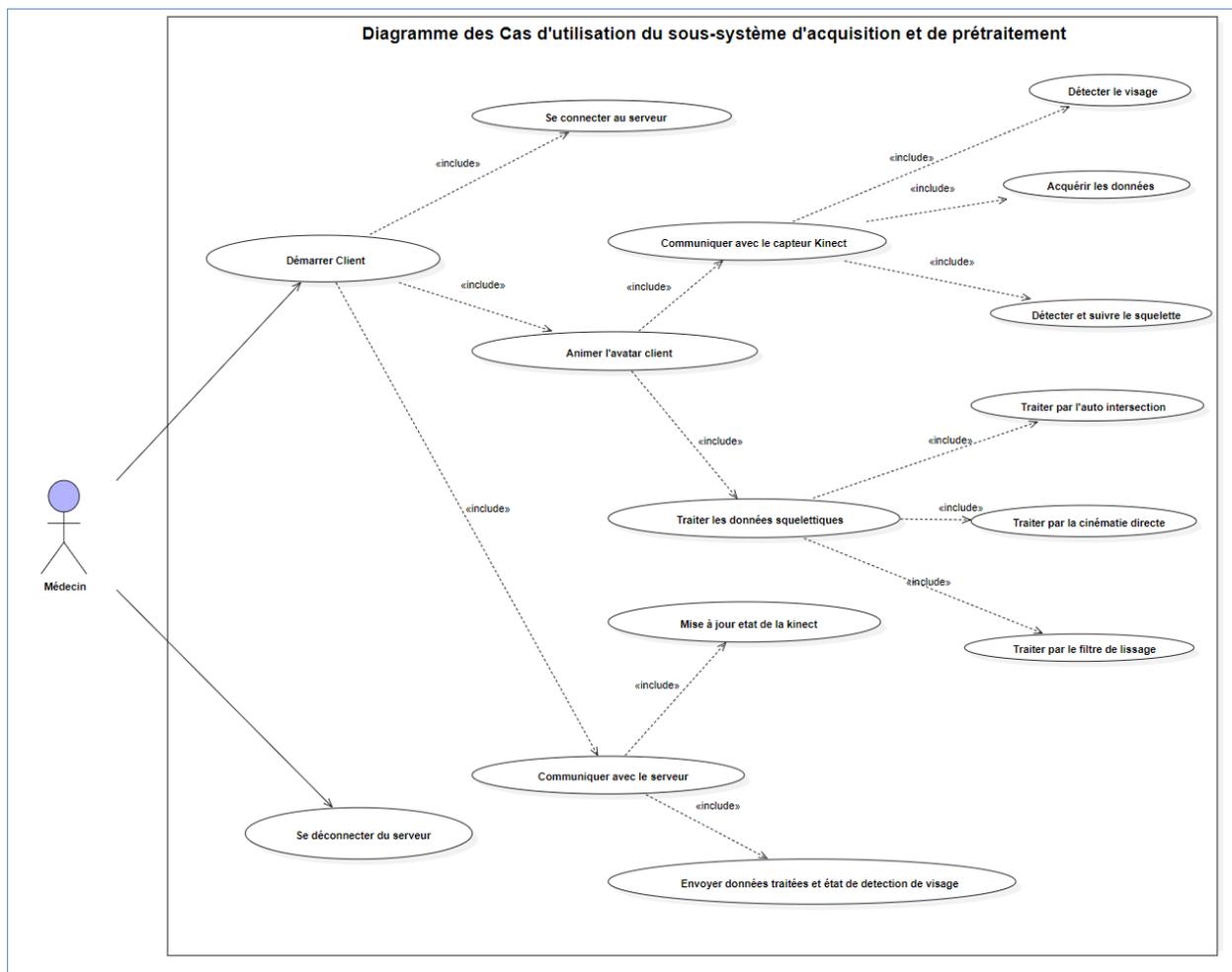


Figure 3.4 : Diagramme de cas d'utilisation « Sous système d'acquisition et prétraitements »

D'une autre part, la figure 3.5 résume les principaux cas d'utilisation réalisés par le médecin/opérateur avec le sous-système animation par fusion :

- **La gestion des patients :** En spécifiant un ensemble d'informations, le médecin/opérateur peut ajouter et supprimer des patients ou modifier leurs informations.
- **Le calibrage des capteurs :** Avant la séance d'examen le médecin/opérateur doit calibrer le système en demandant au patient de se présenter entre chaque couple des capteurs. Une fois les données des clients sont recueillies, le serveur exécute une procédure mathématique qui permet de calculer les matrices de transformation entre tous les couples de capteurs. Ces dernières aident à l'estimation des données squelettiques dans le repère de base.
- **L'ajout d'un examen :** Lors de la visite, le médecin examine l'état de la marche du patient pour suivre l'évolution de sa situation en lui demandant de faire une série d'exercices et de mouvements qui seront capturés pour être animés virtuellement par un Avatar en temps réel. Pour atteindre cet objectif le serveur récolte les différentes données envoyées par les clients, puis il détermine le capteur (repère) de base selon l'état de détection du visage et le nombre des joints traqués dans chaque squelette. Il utilise les matrices de transformation calculées dans la phase de calibrage pour faire l'estimation et la correction des données squelettiques. Par la suite ces données seront fusionnées et adaptées au squelette de l'Avatar pour être animées. Cette animation sera sauvegardée dans la base de données avec les notes et les remarques du médecin.
- **Consultation d'examen par l'animation virtuelle de l'Avatar (animation différée)**
Après avoir sauvegardé les animations et les remarques d'une séance d'examen, le médecin peut consulter virtuellement en mode différé la marche et les mouvements exercés par le patient en réadaptant les données squelettiques enregistrés au squelette de l'Avatar. Ce qui permet d'avoir une animation robuste et réaliste dans la scène virtuelle qui aide le médecin de suivre l'évolution de la situation du patient après chaque séance de rééducation.

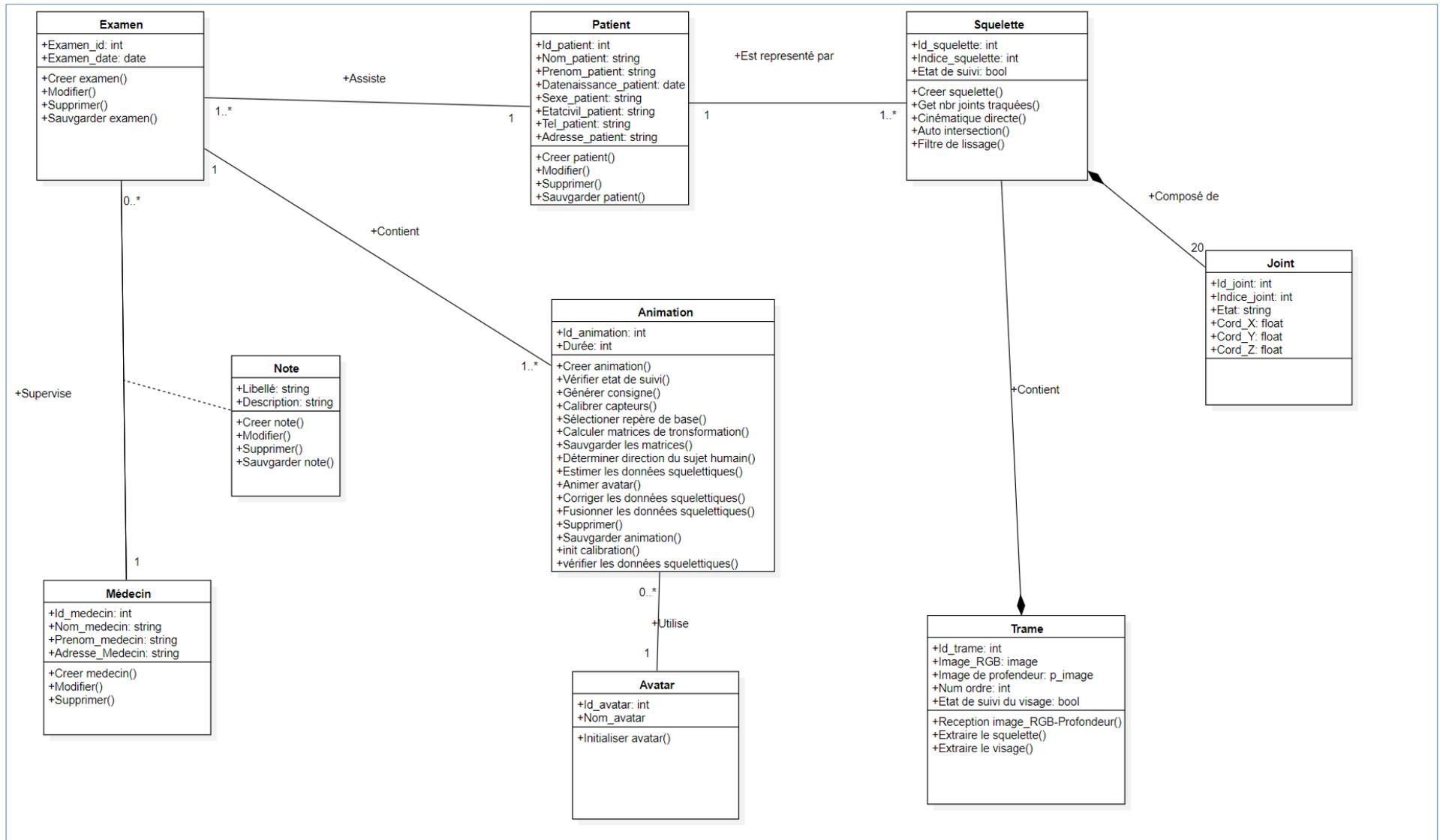


Figure 3.6 : Diagramme de classes

3.3 Diagrammes de séquence

Afin d'afficher les événements par ordre chronologique ainsi que les échanges de messages entre les différentes parties de notre système, nous utilisons les diagrammes de séquence qui nous permettent d'avoir une vue détaillée sur toutes les interactions lors de l'exécution d'un cas d'utilisation.

Nous avons effectué les diagrammes de séquences pour les interactions suivantes :

- Ajout patient, représenté sur la figure 3.7
- Calibrer capteur, représenté sur la figure 3.8
- Animation par fusion, représenté sur la figure 3.9
- Ajout examen, représenté sur la figure 3.10

Cela nous permet d'examiner le comportement des objets et les modifications de leurs états à la suite des réceptions de messages.

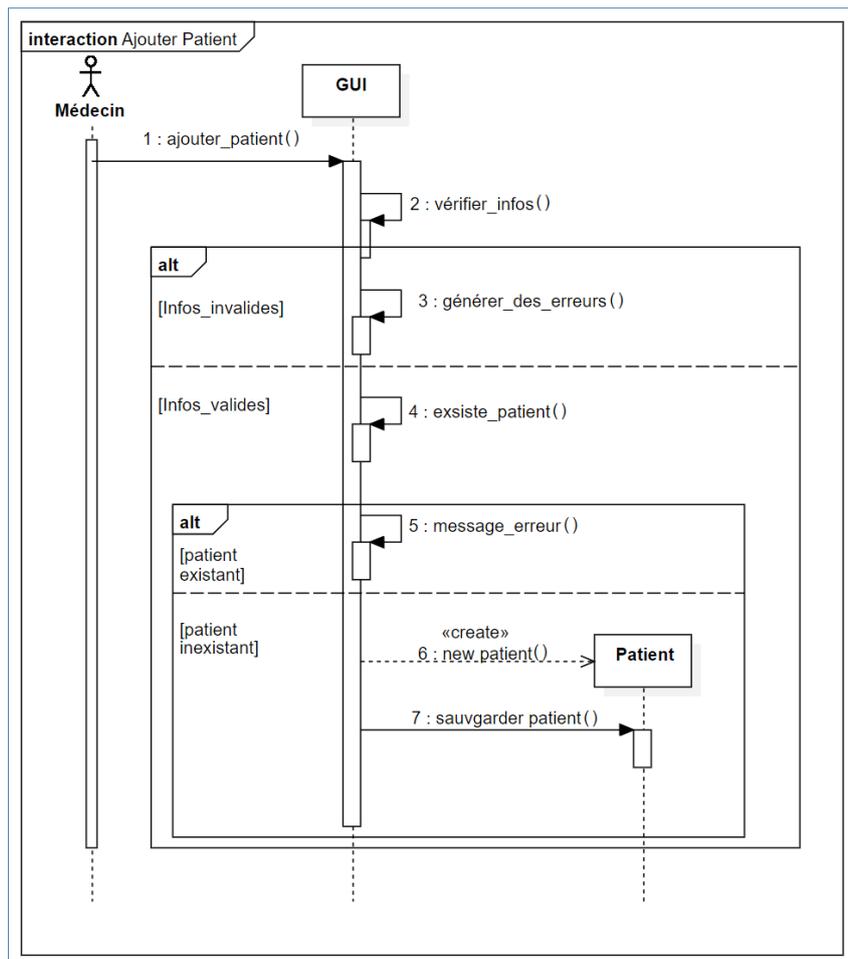


Figure 3.7 : Diagramme de séquence – Ajouter patient

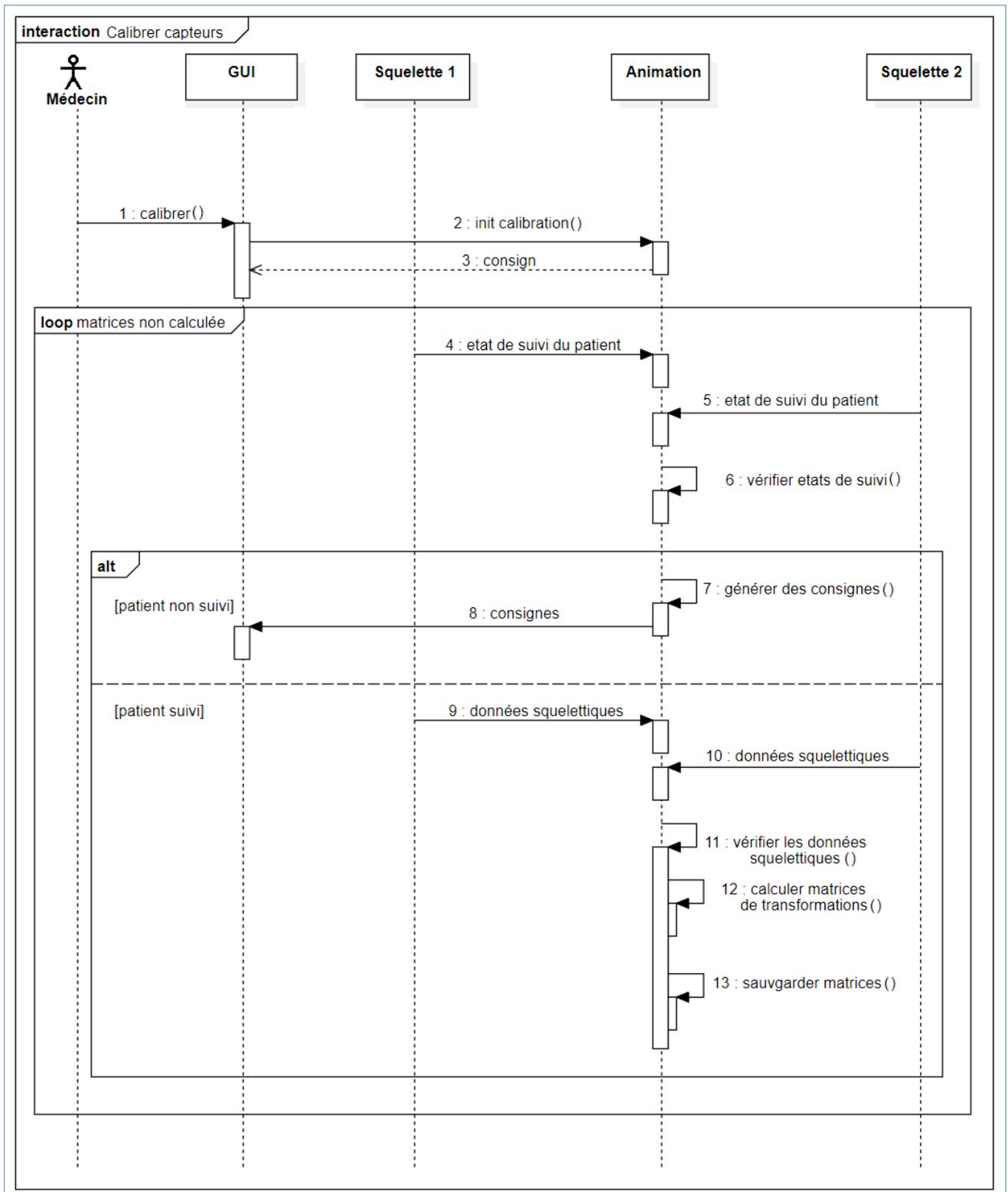


Figure 3.8 : Diagramme de séquence – Calibrer capteurs

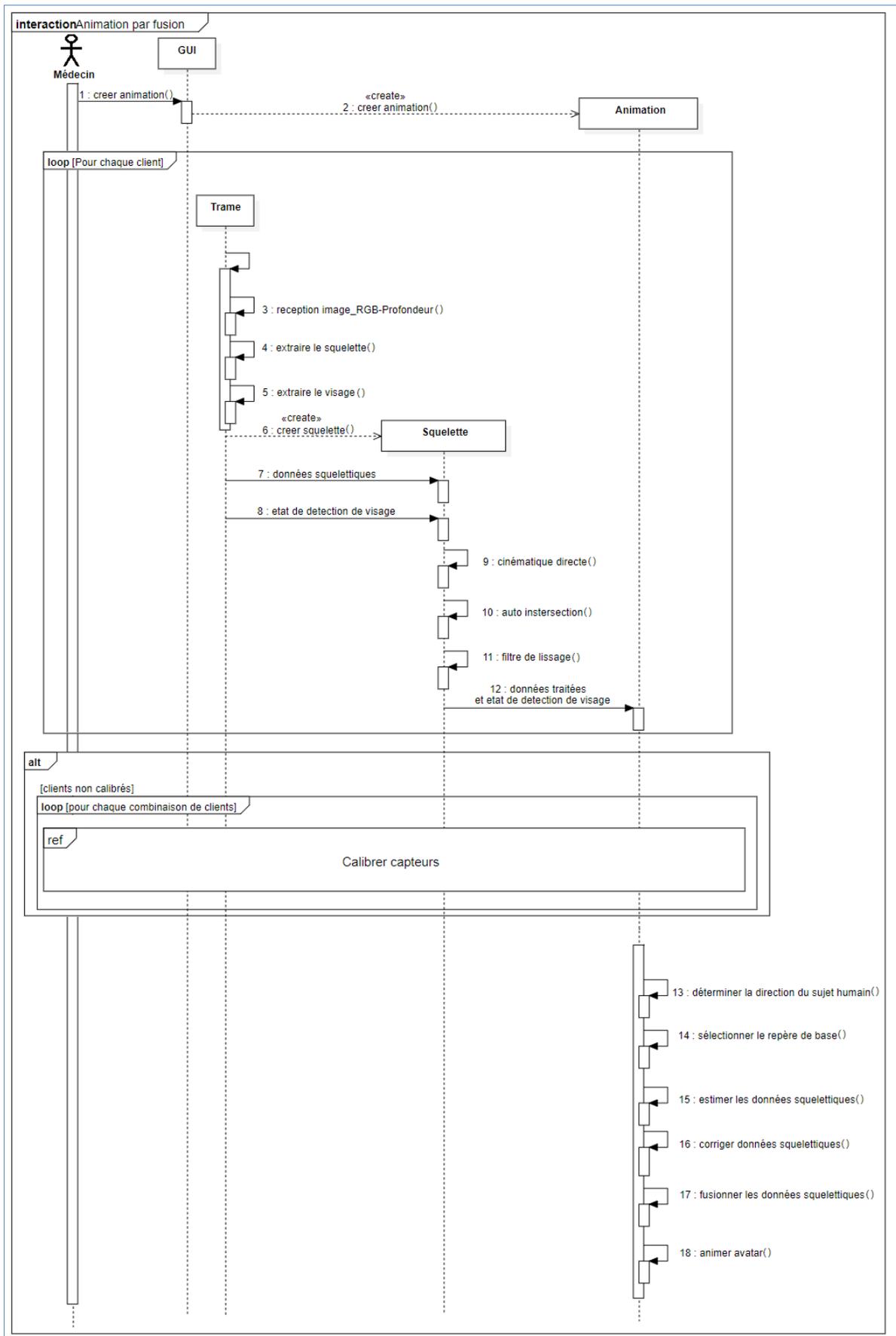


Figure 3.9 : Diagramme de séquence – Animation par fusion

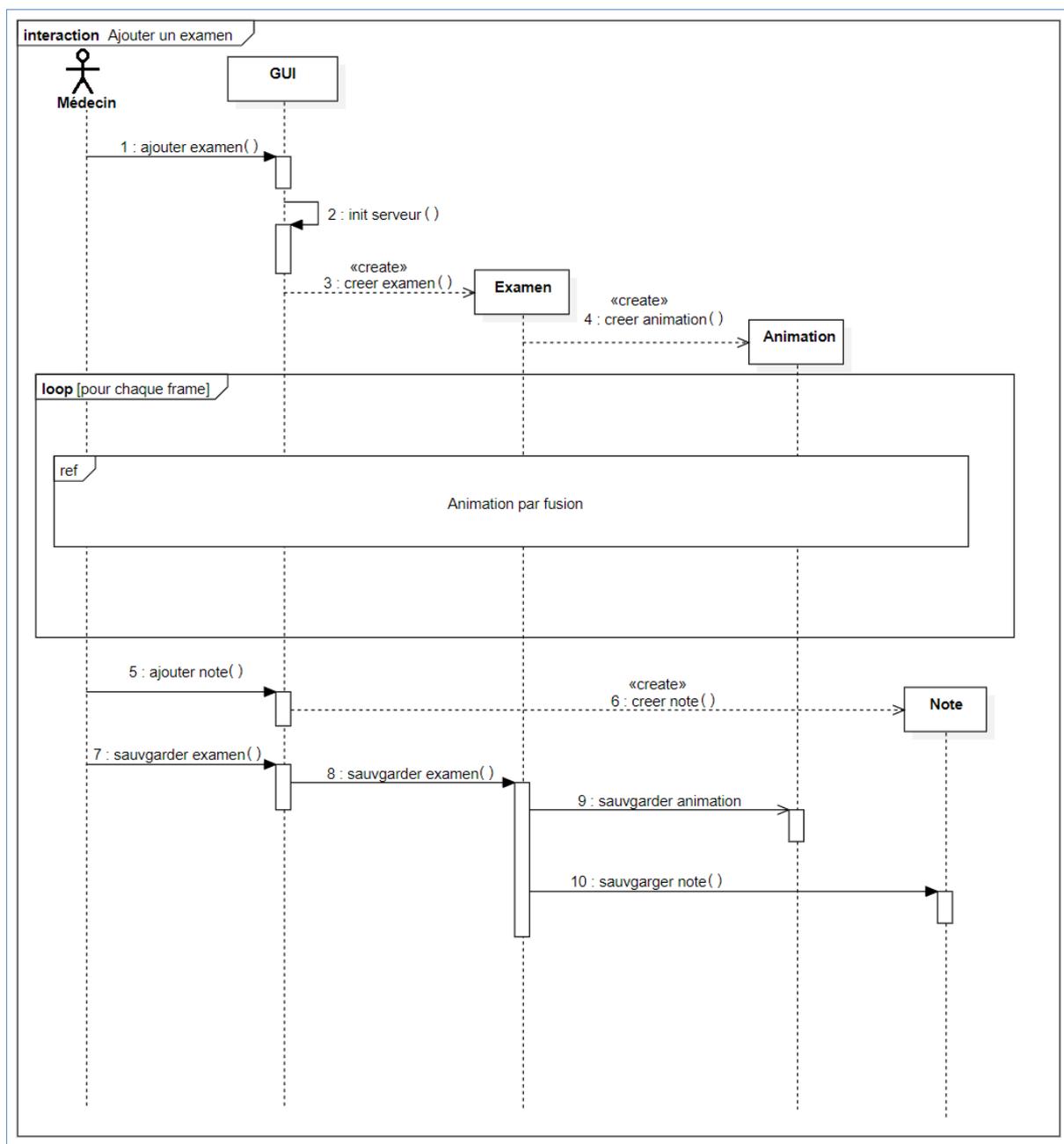


Figure 3.10 : Diagramme de séquence – Ajouter examen

3.4 Diagramme de déploiement

Le diagramme de déploiement montre la configuration physique des différents matériels qui participent à l'exécution de notre système, ainsi que la façon dont les composants d'exploitation y sont installés et la coopération entre eux.

Comme montre la figure 3.11, notre système est constitué de trois nœuds connectés par des liens physiques. Le nœud « capteur Kinect » permet l'acquisition des images RGB et profondeur via ses caméras RGB et infrarouge. Ces images sont transmises via le protocole

USB au nœud « Client » qui les utilise pour l'extraction du squelette et du visage. Le client applique une série de traitements sur ces données pour augmenter la précision et le réalisme de l'animation de l'Avatar.

Par la suite, le nœud « Serveur » reçoit les données squelettiques traitées, par les différents clients, via le protocole TCP/IP. Il applique un ensemble d'étapes citées dans la section 2 afin d'avoir une animation fluide et réaliste de l'Avatar dans le monde virtuel avec une couverture de 360°.

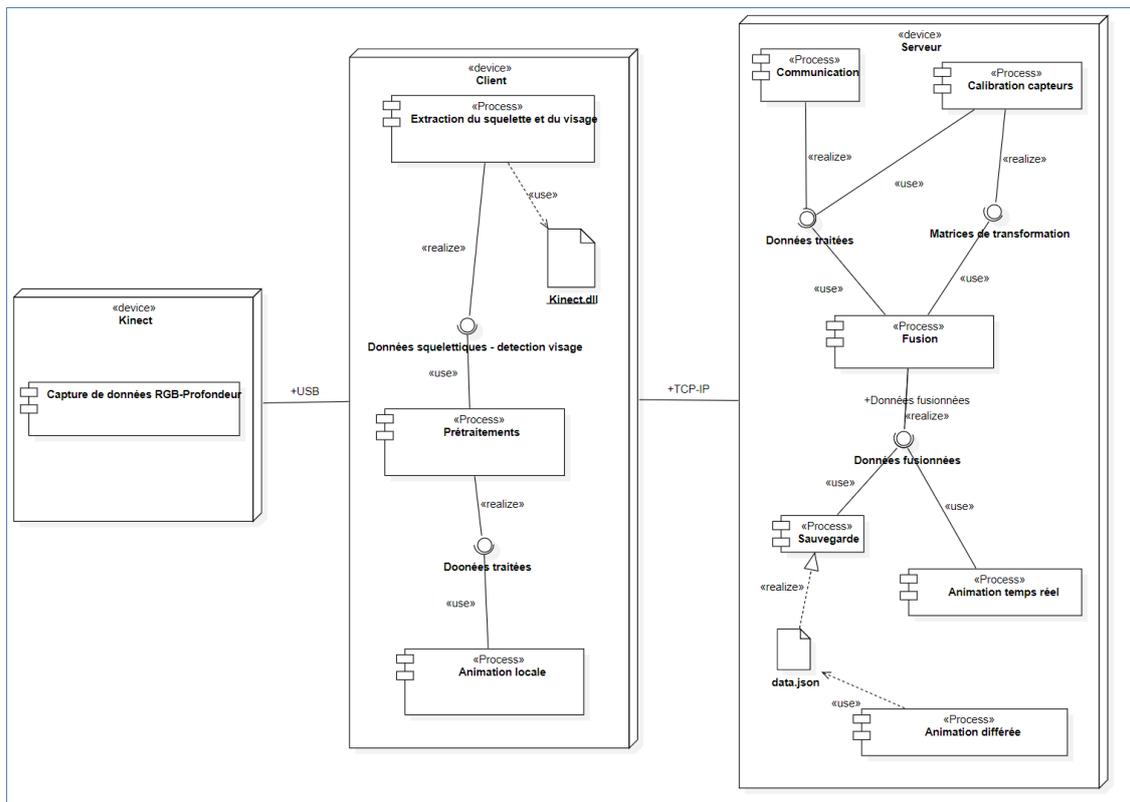


Figure 3.11 : Diagramme de déploiement

4 Installation et configuration matérielle

Nous avons utilisé une architecture distribuée fondée sur la communication Client/serveur pour assurer le transfert des données squelettiques des clients au serveur. Ce transfert repose sur le protocole TCP/IP qui permet d'assurer la communication à distance en temps réel.

Notre système d'acquisition est conçu de telle sorte à être extensible et flexible, c'est à dire que le nombre de capteurs n'est pas fixé préalablement au départ, mais il est plutôt déterminé à partir du nombre de clients connectés lors de l'installation et de la configuration du système.

Ce qui nous donne une liberté d'ajouter d'autres capteurs et un contrôle total des angles de vues des mouvements en respectant une couverture à 360° du sujet humain.

Pour notre expérimentation, nous avons utilisé cinq capteurs Kinect disposés uniformément en périphérie de la zone de travail observée par les capteurs. Comme le montre la figure 3.13, cette disposition forme un pentagone comportant plusieurs points de vue dirigés vers le centre de l'espace de travail. La hauteur par rapport au sol des capteurs est fixée dans notre cas à un (1) mètre (voir la figure 3.12) leurs orientations sont choisies de manière à obtenir un champ de vision vertical supérieur qui est parallèle au sol de telle sorte que le sujet humain doit être vu au complet par chaque capteur.

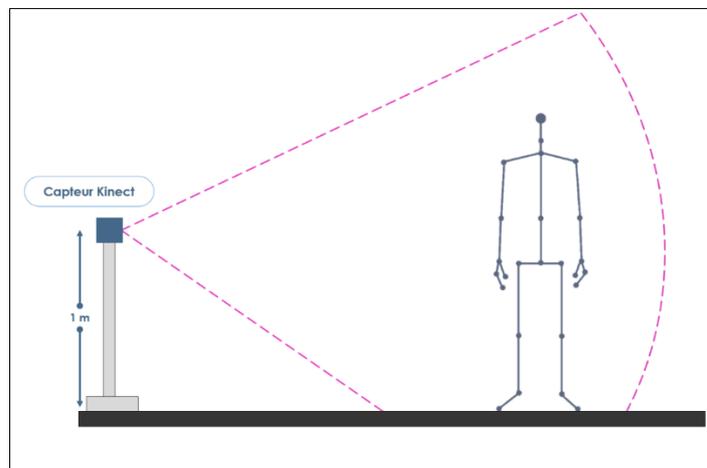


Figure 3.12 : Vue de profil de la position sujet humain/capteur Kinect

Notre méthode de fusion de données squelettiques est robuste aux occlusions ce qui élimine la nécessité d'observer toutes les articulations du sujet humain en même temps. L'emplacement des capteurs est flexible et peut être géré en fonction des mouvements que nous voulons faire suivre par les capteurs (déplacements dans l'espace de travail, rotations du sujet, poses fixes, etc.). Le sujet humain peut donc sortir de la zone observée d'un ou plusieurs capteurs sans empêcher l'estimation de la position des joints du squelette final fusionné.

Le schéma global de la configuration matérielle du système proposé est représenté sur la figure 3.13.

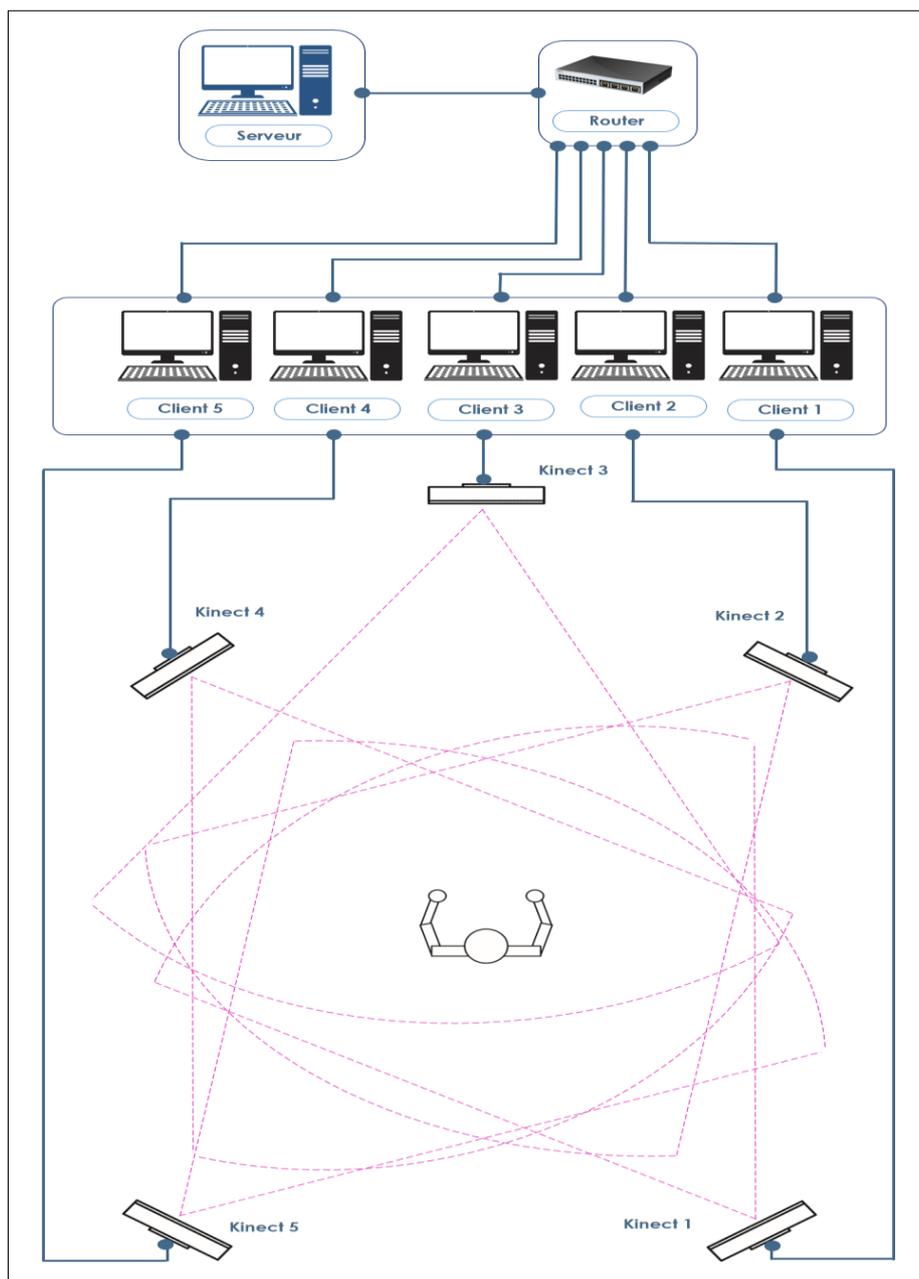


Figure 3.13 : Schéma global de la configuration matérielle du système proposé

5 Processus de l'animation virtuelle de l'Avatar

5.1 Acquisition et prétraitements de données squelettiques brutes

Le module de capture de mouvement utilisé dans notre application est basé sur les capteurs Kinect à bas coût que Microsoft a mis à la disposition du grand public depuis 2010. Notre choix est justifié par les divers avantages apportés par ces capteurs Kinect. Ils fournissent des fonctionnalités plus riches et variées par rapport aux autres systèmes de capture de mouvement existants. Ces capteurs offrent un suivi fiable du squelette humain, grâce à l'algorithme de

Shotton *et al.* qui permet d'extraire le squelette en temps réel à partir des images de profondeurs. De plus, par le biais de leur projecteur infrarouge, qui diffuse une lumière structurée formant un nuage de point projetés sur les surfaces de la scène, le capteur n'est pas influencé par les textures se trouvant dans l'environnement. Il permet de percevoir des objets non texturés même dans une scène de faible luminosité. De même, pour simplifier leur utilisation dans le développement des applications, Microsoft offre une multitude de bibliothèques telle que le Microsoft SDK Kinect pour Windows qui est un kit de développement logiciel (*Software Development Kit*). En plus de sa facilité d'installation, cet environnement de développement contient des pilotes et des API très riches.

L'utilisation des capteurs Kinect dans notre système nous permet de recevoir avec une cadence de 30 frames par seconde les images de couleur et de profondeur de la scène observée, qui sont utilisées dans les différentes phases de traitements de ce module, à savoir : l'extraction du squelette, les prétraitements et la détection du visage.

5.1.1 Extraction du squelette

Afin d'exploiter et manipuler plus aisément les données fournies par le capteur Kinect de chaque machine client, il est nécessaire de les transformer en modèle qui se rapproche de la structure de l'être humain. Pour cela le SDK Kinect intègre l'algorithme de Shotton *et al.* Celui-ci permet de reconnaître la posture humaine à partir d'une image de profondeur et de fournir un squelette de 20 articulations reliées entre elles par des liens rigides représentant les os en formant un arbre hiérarchique dont la racine est le bassin. Un ensemble d'APIs nous fournissent la position 3D et l'état de suivi de chaque articulation. Trois états de suivi d'une articulation peuvent être distingués :

- Traqué : Si l'articulation est bien suivie par le capteur Kinect, la position fournie est correcte.
- Non Traqué : Si l'articulation n'est pas visible par le capteur, la position fournie est fausse.
- Inféré : Si l'articulation est occultée par une autre partie du corps, sa position est déduite à partir de la position d'une autre articulation.

Chaque articulation du modèle squelettique est identifiée par son nom comme le montre la figure 3.14.

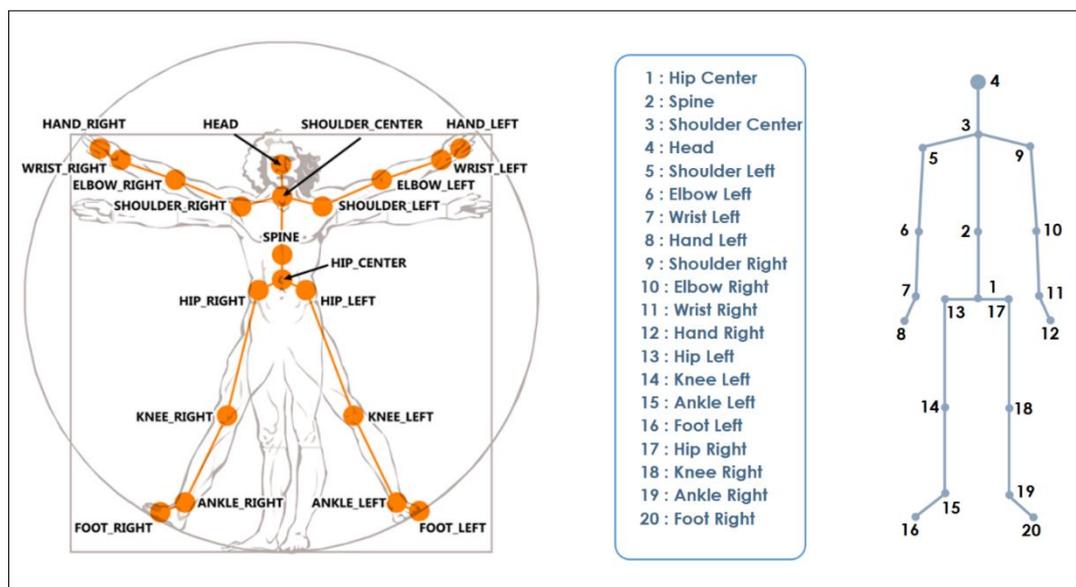


Figure 3.14 : Représentation du modèle squelettique envoyées par la Kinect

5.1.2 Prétraitement des données squelettiques

Afin de garantir le réalisme de l'animation dans le monde virtuel des mouvements exercés par le sujet humain et d'évaluer l'exactitude de notre modèle final, nous ajoutons des ajustements sur ses articulations en essayant de rectifier les positions ayant des valeurs erronées ou inférées. Ce processus revient aux bruits contenus dans les images de profondeur qui influent sur la précision de l'informations sur les positions des articulations du modèle squelettique brut récupéré par le capteur Kinect.

Pour limiter ces erreurs, les données squelettiques acquises du capteur de chaque client sont soumises localement à une série de traitements avant d'être transmises au serveur. Ces traitements sont inspirés des travaux réalisés dans [66] et modifiés selon notre cas, à savoir :

5.1.2.1 Auto-intersection

Le problème d'auto-intersection se manifeste clairement lorsque la personne colle ses mains ou ses poignets à son buste. Au niveau de l'Avatar, les articulations participantes, des extrémités des membres supérieurs, au mouvement vont être à l'intérieur de l'Avatar (elles traversent le buste) et non pas collés à sa surface comme les gestes exercés par le sujet humain dans le monde réel. Pour résoudre ce problème, l'auteur dans [66] a créé deux boîtes englobantes invisibles autour du buste afin de garder les articulations à une certaine distance. Ces boîtes sont définies par des seuils de distance sur les trois axes (X, Y, Z). A chaque frame,

l'auteur calcule la distance entre les articulations des extrémités des membres supérieurs et l'articulation du bassin ou l'articulation du cou pour détecter l'auto-intersection. Si les trois distances (D_x , D_y , D_z) sont inférieures aux seuils définis ($Seuil_x$, $Seuil_y$, $Seuil_z$), alors une intersection est détectée. Il effectue par la suite une correction en repoussant l'articulation qui traverse la boîte englobante par un vecteur de translation. Ce dernier relie l'articulation intersectée avec l'articulation la plus proche (le bassin ou le cou) dont la longueur du vecteur est la distance entre eux (Voir Figure 3.15).

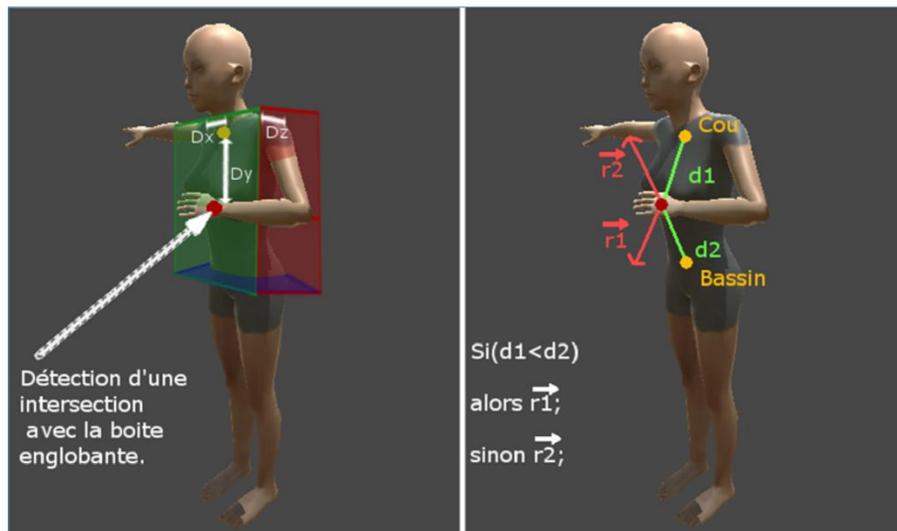


Figure 3.15 : Détection et correction de l'auto intersection

Le pseudo algorithme 3.1 résume le principe du traitement de l'auto-intersection :

Algorithme 3.1: Correction de l'auto-intersection des articulations [66]

Entrée:

- Articulations concernées par l'auto-intersection.

Début:

Pour (chaque articulation concernée par l'auto intersection) faire

VecteurArticulationBassin = ArticulationPos - BassinPos;

VecteurArticulationCou = ArticulationPos - CouPos;

Si(LongueurVecteurArticulationBassin < LongueurVecteurArticulationCou) Alors
 ArticulationPos = ArticulationPos + VecteurArticulationBassin;

Sinon

 ArticulationPos = ArticulationPos + VecteurArticulationCou;

Fsi;

Fait;

Fin.

Sortie: Positions des articulations corrigées.

5.1.2.2 Cinématique directe

Pour gérer le problème des valeurs imprécises fournis par le capteur Kinect sur les positions des articulations des extrémités (les pieds, les mains et la tête), lorsque celles-ci sont inférées et non détectées, nous appliquons le traitement de la cinématique directe qui consiste à calculer la position de l'articulation inférée (et/ou non détectée) en se basant sur la longueur de l'os, la rotation de l'os parent et la position de l'articulation parente. Après l'identification des articulations inférées par le capteur, nous appliquons une procédure de calcul mathématique permettant de déterminer leurs positions en se basant sur la formule suivante [66] :

$$Pos_i = Pos_{i-1} + \left(\frac{Pos_{i-1} - Pos_{i-2}}{|Pos_{i-1} - Pos_{i-2}|} \right) * l_{i-1} \quad (3.1)$$

1. Pos_i : La position de l'articulation extrémité à corriger.
2. Pos_{i-1} : La position de l'articulation parente de l'extrémité à corriger.
3. l_{i-1} : Longueur du lien reliant l'articulation extrémité à son articulation parente.
4. $\left(\frac{Pos_{i-1} - Pos_{i-2}}{|Pos_{i-1} - Pos_{i-2}|} \right)$ Vecteur de direction normalisé du lien $i - 2$.

La figure 3.16 montre un exemple d'application de la cinématique directe sur la position de l'articulation inférée de la main gauche en lui affectant la nouvelle position calculée.

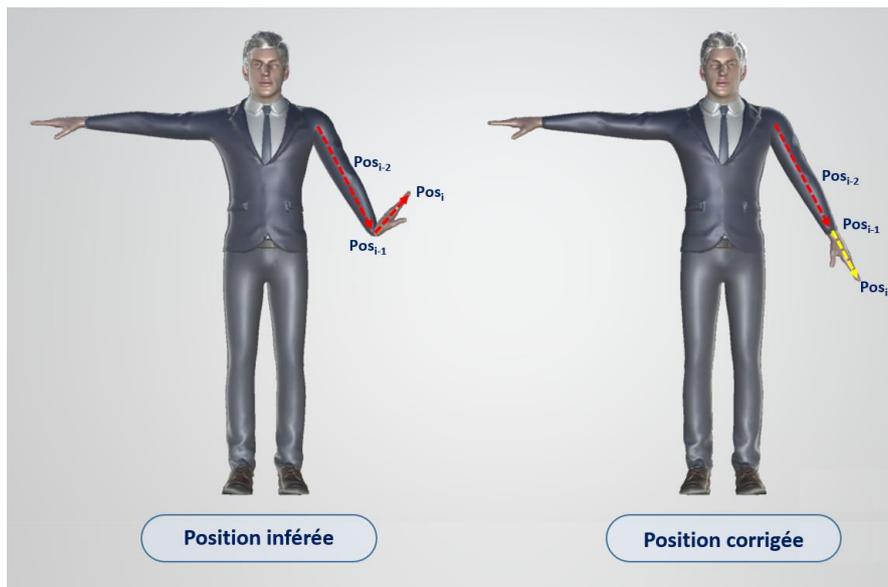


Figure 3.16 : Exemple de cinématique directe appliquée à la main

5.1.2.3 Filtre de lissage

Le filtre de lissage prend en considération l'aspect temporel pour ajuster les positions des articulations. Il permet de lisser les données squelettiques du frame courant en fonction des frames précédents par le calcul de la moyenne mobile dont son expression est la suivante [66] :

$$F_k = \frac{1}{n+1} \left(\sum_{i=k-n}^{k-1} F_i + X_k \right), K > n \quad (3.2)$$

Avec :

F_k : La nouvelle position filtrée de l'articulation dans le frame courant.

n : La taille de la fenêtre du filtre.

F_i : Les positions filtrées de l'articulation dans les frames précédents.

X_k : Valeur brute de la position de l'articulation dans le frame courant.

Pour déterminer la position ajustée de chaque articulation du squelette, nous calculons d'abord la moyenne de ses positions filtrées sur les n derniers frames, en leur ajoutant sa position dans le frame courant. Puis, nous assurons que la nouvelle position n'est pas très éloignée de la position fournie par le squelette, en calculant la distance entre les deux positions. Si la déviation dépasse un seuil défini, nous réajustons la position en réduisant la distance, grâce à la formule suivante [66] :

$$F_k = F_k * \left(\frac{\tau}{\Delta_k} \right) + X_k * \left(1 - \frac{\tau}{\Delta_k} \right) \quad (3.3)$$

tel que :

τ : Seuil maximal de la déviation = 0.2 m. Valeur choisie suite à des tests.

Δ_k : Distance entre la position filtrée et la position brute de l'articulation dans le frame k .

5.1.3 Détection de visage

Pour déterminer l'orientation du sujet humain par rapport à l'ensemble des capteurs Kinect du réseau dans le monde réel, nous avons déployé la fonctionnalité de la détection et le suivi du visage du sujet humain dans notre système. Elle est basée sur le travail proposé par Nikolai *et al.* dans [67] qui est adopté par le SDK de Microsoft Kinect. Leur travail utilise à la fois les données RGB et de profondeur pour la détection et le suivi du visage humain en 3D. Ils se basent sur les modèles d'apparence active (AAM) qui servent à adapter un modèle génératif

de forme et d'apparence d'un objet à une image. L'AAM (*Active Appearance Model*) leur permet de suivre en 2D le visage du sujet humain en temps réel et de manière précise. À l'aide d'un détecteur de visage, ils commencent à initialiser leur système de suivi de visage en trouvant un rectangle de visage dans une image de couleur RGB à l'aide d'un détecteur de visage. Puis, ils utilisent un réseau de neurones pour trouver cinq points à l'intérieur de la zone du visage (le centre des yeux, les coins de la bouche et le bout du nez). Ensuite, ils calculent les dimensions du visage suivi à partir de ces cinq points et lui mettent leur modèle de visage 3D de manière appropriée. L'avantage principal de cet algorithme est qu'il offre pour chaque visage détecté la position 2D et 3D de la tête ainsi que sa rotation. La figure 3.17 nous montre un exemple de détection du visage en projetant un masque 3D sur le visage du sujet humain dans l'image RGB.

Malgré l'avantage de fonctionner en temps réel et de manière robuste, la précision de la méthode de détection du visage peut être affectée par plusieurs facteurs, et qui sont :

- 1) La luminosité : le visage du sujet humain doit être bien éclairé sans trop de nuances dures. Un éclairage arrière ou latéral intense peut rendre le suivi plus difficile.
- 2) La distance par rapport au capteur Kinect : plus le sujet humain est plus proche du capteur, le suivi sera meilleur. À plus courte distance, les données de profondeur de la Kinect sont plus précises et la fonctionnalité de suivi du visage peut donc calculer les points 3D du visage avec plus de précision.
- 3) L'occultation : le sujet humain ne doit pas porter des lunettes épaisses ni une grande barbe, pour avoir un meilleur suivi de visage.

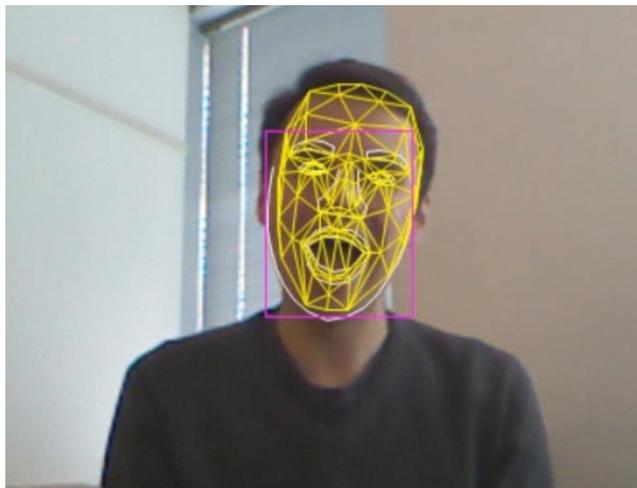


Figure 3.17 : Exemple de suivi du visage fournit par le SDK [67]

Après l'acquisition des données RGB et profondeurs utilisées pour l'extraction du squelette et la détection du visage du sujet humain, ainsi que les différents traitements subit sur ces derniers, chaque client prépare le modèle de données contenant son identité et les données squelettiques corrigées avec l'état de détection du visage pour être transférer au serveur.

5.2 Calibrage

Avant de fusionner les données squelettiques provenant des différents clients, il est nécessaire qu'elles soient toutes représentées dans un même repère de coordonnées. Ce repère peut être n'importe lequel des systèmes de coordonnées du capteur noté comme repère globale. Chacun des systèmes doit donc être calibré sur ce dernier, c'est-à-dire que nous devons calculer la matrice de transformation rigide effectuant la translation des données squelettiques du repère local vers le repère global.

5.2.1 Représentation d'une articulation

Le modèle squelettique donné par le capteur est composé d'un ensemble d'articulations. Chacune d'elles est représentée par un point P de coordonnées cartésiennes p_x, p_y, p_z (figure 3.18). Nous représentons alors ses coordonnées par le vecteur :

$$P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

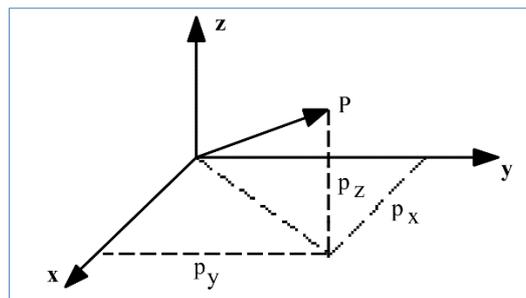


Figure 3.18 : Représentation d'une articulation dans le repère capteur

5.2.2 Changement de repère

Comme notre système est composé de plusieurs capteurs Kinect, chacun d'eux possède son propre point de vue qui constitue son repère de représentation des données squelettiques. Or, les données envoyées au serveur par les différents clients sont représentées chacune dans des repères distincts. Cela nous mène à élaborer une méthode permettant de faire des

conversions des données squelettiques d'un repère à un autre et de les transformer en un seul repère global (changement de repère) afin de pouvoir les fusionner correctement.

Pour arriver à cette fin, nous devons calculer les matrices de transformation permettant d'élaborer la correspondance et la relation entre les différents repères. Nous avons utilisé un calibrage fort basé sur les données des positions des vingt articulations du modèle squelettique.

Comme une même articulation P est capturée en même temps par deux capteurs de repère R_i et R_j respectivement (voir figure 3.19), nous pouvons avoir deux positions de ce même point dans les deux repères : $P_i = (x_i, y_i, z_i)$ et $P_j = (x_j, y_j, z_j)$. D'où, nous pouvons exprimer la relation entre eux par l'équation suivante :

$$P_i = M_{ij}P_j \quad (3.4)$$

tel que :

P_i : Cordonnées du point P dans le repère R_i .

P_j : Cordonnées du point P dans le repère R_j .

M_{ij} : Matrice de transformation 3x3 entre le repère R_i et le repère R_j .

Et plus précisément :

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} \quad (3.5)$$

Tel que : $a, b, c, d, e, f, g, h, i$ sont des valeurs de type réel.

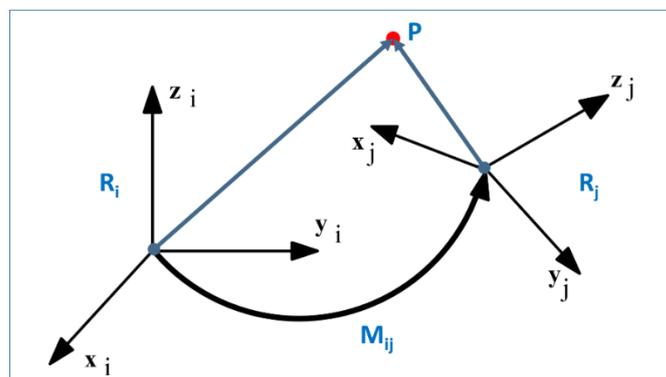


Figure 3.19 : Transformation de repère

De l'équation (3.5), on obtient le système d'équation suivant :

$$\begin{cases} x_i = x_j \times a + y_j \times b + z_j \times c \\ y_i = x_j \times d + y_j \times e + z_j \times f \\ z_i = x_j \times g + y_j \times h + z_j \times i \end{cases} \quad (3.6)$$

En généralisant le système équation (3.6) aux vingt articulations du squelette, nous obtenons :

$$\begin{bmatrix} x_{1j} & y_{1j} & z_{1j} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{1j} & y_{1j} & z_{1j} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{1j} & y_{1j} & z_{1j} \\ x_{2j} & y_{2j} & z_{2j} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{2j} & y_{2j} & z_{2j} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{2j} & y_{2j} & z_{2j} \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ x_{20j} & y_{20j} & z_{20j} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{20j} & y_{20j} & z_{20j} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{20j} & y_{20j} & z_{20j} \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{bmatrix} = \begin{bmatrix} x_{1i} \\ y_{1i} \\ z_{1i} \\ x_{2i} \\ y_{2i} \\ z_{2i} \\ \vdots \\ \vdots \\ \vdots \\ x_{20i} \\ y_{20i} \\ z_{20i} \end{bmatrix} \quad (3.7)$$

Qui peut être représentée sous la forme de : $A \times X = B$ (3.8)

A : matrice des articulations P_{ki} dans le repère R_i de taille 60 x 9

B : vecteur des articulations P_{kj} dans le repère R_j de taille 60

X : Vecteur des coefficients de la matrice de transformation de taille 9

k : entier $0 < k \leq 20$

Pour calculer les facteurs de la matrice de transformation M_{ij} (vecteur X), nous simplifions l'équation (3.8) en :

$$A \times X = B \quad (3.8)$$

$$A^T \times A \times X = A^T \times B \quad (3.9)$$

$$(A^T \times A)^{-1} \times (A^T \times A) \times X = (A^T \times A)^{-1} \times (A^T \times B) \quad (3.10)$$

On obtient : $X = (A^T \times A)^{-1} \times A^T \times B$ (3.11)

Tel que : A^T : la matrice transposée de A

Après l'utilisation de la formule (3.11), nous obtenons un vecteur de neuf éléments. Nous le transformons en matrice de 3x3 qui représente la matrice de transformation M_{ij} .

De ce fait, nous sommes arrivés à trouver la correspondance entre les deux repère R_i et R_j . Cependant, notre système est constitué de plusieurs capteurs, pour réaliser le calibrage dans notre cas, nous avons utilisé une technique permettant de faciliter et d'optimiser les calculs des matrices entre les capteurs. Tout d'abord nous demandons au sujet humain de se placer en face de la première combinaison des capteurs (C_1, C_2) (étape 1), s'il n'est pas bien traqué par les deux capteurs alors un ensemble de consignes seront afficher pour le guider, sinon nous calculons les matrices de transformation rigides comme suit : $C_1 = M_{12}C_2$, $C_2 = M_{21}C_1$ avec $M_{21} = (M_{12})^{-1}$ (inverser une matrice de calibration changera la direction de la transformation). Par la suite, nous appliquons la même procédure avec les autres combinaisons des capteurs (C_2, C_3), (C_3, C_4), (C_4, C_5) comme le détaille la figure 3.20.

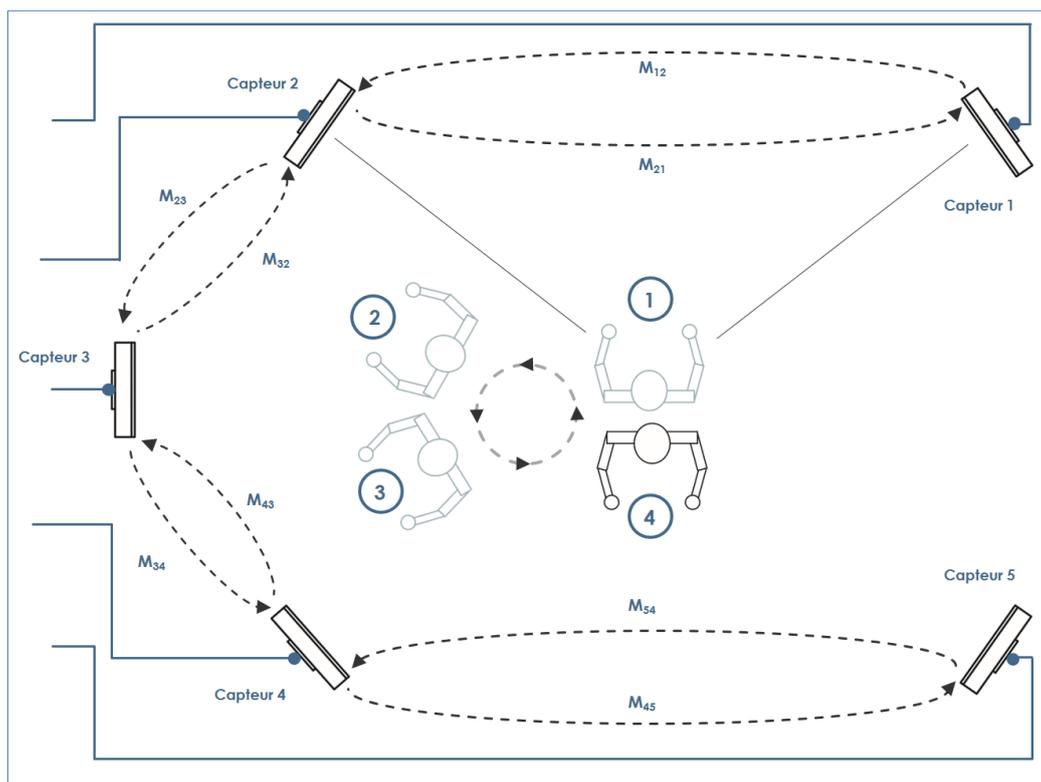


Figure 3.20 : Calibration des capteurs

Une fois ces matrices sont calculées, une autre étape de calibration est nécessaire. Elle consiste à calculer les matrices de transformation des autres combinaisons possibles entre les capteurs du système (par exemple (C_1, C_3), (C_1, C_4), (C_1, C_5), (C_2, C_4) (C_2, C_5) ...). Pour ce faire, nous utilisons les relations de transformations des repères :

$$C_i = M_{ij} \times C_j \quad (3.12)$$

$$C_j = M_{jk} \times C_k \quad (3.13)$$

$$C_i = M_{ij} \times M_{jk} \times C_k \quad (3.14)$$

Par la même logique M_{ik} est directement obtenu si l'on connaît M_{ij} et M_{jk}

$$M_{ik} = M_{ij} \times M_{jk} \quad (3.15)$$

A la fin de ce processus, nous obtenons toutes les matrices de transformation permettant de faire toutes les translations des données entre les repères des capteurs. Elles servent par la suite à la fusion homogène des données squelettiques dans un seul repère appelé repère global.

5.3 Fusion des données squelettiques et animation de l'Avatar

5.3.1 La fusion des données

La fusion des données est le processus de combinaison de données provenant de différents clients. Ces données corrigées et filtrées par le module d'acquisition et de prétraitement peuvent être plus ou moins précises. Notre objectif étant la réduction de l'incertitude des mesures de position des joints du squelette du sujet humain obtenues à partir d'une seule vue par un seul capteur Kinect, en combinant plusieurs mesures en provenant de plusieurs capteurs Kinect disposés de telle sorte à avoir une vue panoramique de 360° du sujet humain. Voir le schéma global de la configuration matérielle de notre système de la Figure 3.13 montrant la disposition spatiale des capteurs Kinect dont nous voulons fusionner les données.

L'extraction du squelette d'un seul point de vue donne un très bon résultat en temps réel, cependant il reste trop limitatif pour notre système. D'abord, le squelette fourni peut avoir une absence d'informations sur les parties cachées ou occultées du corps du sujet humain. Puis, lorsque le sujet humain est vu de profile, le capteur Kinect renvoie une capture aléatoire et irréaliste, de même lorsque le sujet humain est vu de dos par le capteur les mouvements du sujet humain sont permuter et inverser.

Le but principal de l'utilisation de la fusion des données est de réduire l'impact des occlusions en augmentant la couverture spatiale par la redondance de l'information et de garantir la fiabilité et la précision du système de suivi de mouvement ce qui mène à un squelette fusionné complet, pertinent, robuste et précis.

5.3.1.1 Acquisition des données et identification des clients

Pour un transfert fiable et stable des données squelettiques entre les clients et le serveur, à chaque frame ces données sont converties en un flux d'octets avant leurs transferts. Une première étape consiste d'abord à convertir les objets contenant ces données en une représentation JSON tout en conservant leurs caractéristiques et leurs types initiaux. Puis, un processus de sérialisation leur est appliqué. Il convertit le résultat JSON en un flux d'octets pour les envoyer au serveur. Dès leur réception, le processus inverse est appliqué qui représente la désérialisation en représentation JSON, suivi par la conversion en objets pour les rendre à leur état initial.

Afin d'exploiter les données squelettiques envoyées par les différents clients, nous devons d'abord les identifier dès leur réception à chaque frame. Ce qui permet de distinguer entre eux et de faciliter leur utilisation et leur manipulation par le serveur. Cela est fait grâce à l'identifiant timbré du client émetteur sur chaque paquet transféré.

En se basant sur le nombre d'utilisateurs connectés au sein de notre réseau, nous indexons les machines clientes par une méthode d'indexation incrémentale offrant à chacun un identifiant unique dès sa connexion au serveur.

5.3.1.2 Sélection du squelette de base ou de référence

La précision et la fiabilité des données squelettiques fournies par chaque capteur augmentent lorsque le sujet humain se met face au dispositif et diminuent lorsqu'il est capturé de profil. Pour cela, nous avons été menés à choisir pour chaque frame le capteur de base (ou de référence) fournissant la meilleure représentation des données squelettiques du sujet humain, appelée dans notre cas le squelette de base. Nous avons affecté la tâche de sélection du squelette de base au serveur qui se base sur les états de détection du visage de chaque capteur client. Il choisit celui qui renvoie un état de détection du visage positif. Trois cas peuvent être distingués :

- Premièrement, lorsque le visage est détecté par un seul capteur, le squelette renvoyé par ce dernier sera choisi comme squelette de base.
- Deuxièmement, lorsque le visage est détecté par plusieurs capteurs, le squelette ayant le plus d'articulations traquées parmi eux sera considéré comme squelette de base.
- Troisièmement, si aucun des clients n'a détecté de visage, le squelette de base choisi sera celui qui a le plus grand nombre d'articulations suivies/traquées.

5.3.1.3 La fusion des données squelettiques

Après la sélection du squelette de base, nous l'utilisons comme le premier noyau de la construction du squelette final. À chaque frame, nous identifions en premier lieu toutes les articulations non traquées par le capteur de base. Puis, pour chaque articulation non traquée, nous récupérons sa position à partir des autres clients quand elle est bien suivie :

- Si un seul client l'a traquée, alors nous convertissons sa position au repère de base en utilisant les matrices de transformations calculées dans le module de calibrage. Nous affectons par la suite la position convertie au squelette de base.
- Si cette articulation est bien traquée par plusieurs capteurs clients, nous convertissons toutes ces positions de cette même articulation à l'aide des matrices de transformation au repère de base. Cette tâche nous permet de transformer les données squelettiques des repères des capteurs au repère du squelette de base afin d'assurer l'homogénéité des données (représentées dans le même repère) lors de leur fusion. Par la suite, nous appliquons l'algorithme CHA (*Clustering hiérarchique ascendant*), un algorithme de classification hiérarchique ascendant non supervisé permettant de classifier les objets observés dans des classes appelées clusters où les objets de même classe doivent être similaires et les objets de deux classes différentes doivent être distincts. Nous l'appliquons sur les positions converties afin de les regrouper dans des clusters en se basant sur la distance euclidienne dans le but d'avoir un cluster qui regroupe le maximum de positions les plus proches. Cette étape nous permet d'éliminer les positions éloignées de la position exacte de l'articulation et d'augmenter la précision et la fiabilité de la nouvelle position calculée. Nous affectons par la suite la nouvelle position obtenue par la moyenne des articulations du cluster choisi à l'articulation manquante dans le squelette de base. Nous obtenons à la fin un squelette complet représentant le sujet humain dans le monde virtuel.

En dernier lieu, le squelette résultant sera converti vers un repère que nous appelons repère global. Ce dernier est choisi comme le repère du premier capteur connecté au réseau (un choix expérimental). Le squelette de base de chaque frame doit être représenté dans ce repère. Ce qui permet d'avoir une séquence de squelettes homogènes représentés dans le même repère fournissant une animation fluide et stable de l'Avatar lors du changement dans le repère de base.

La figure 3.21 résume l'approche proposée pour la fusion des données squelettiques provenant des différents clients.

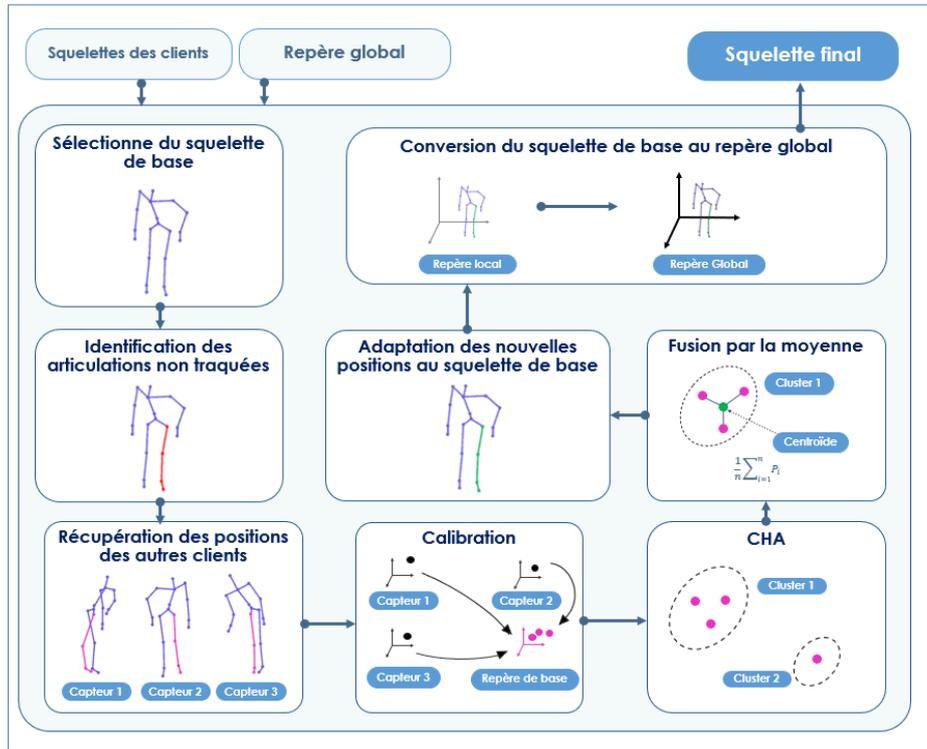


Figure 3.21 : Approche proposée pour la fusion des données squelettiques

5.3.2 Processus d'animation de l'Avatar

Dans cette partie, nous allons détailler les différentes étapes qui constituent le module de l'animation de l'Avatar dans un monde virtuel. Nous allons expliquer comment nous contrôlons et animons notre Avatar à chaque frame ainsi que la reproduction des mouvements exercés par le sujet humain dans le monde virtuel en temps réel.

5.3.2.1 Transformation des données squelettiques

Après l'obtention du modèle squelettique fusionné contenant l'ensemble des positions de toutes les articulations, nous utilisons une méthode qui consiste à transformer les positions résultantes en des angles de rotation, et ceci pour permettre d'avoir une animation fluide et réaliste des parties du corps de l'Avatar et aussi c'est la seule manière de contrôler l'Avatar.

Pour calculer les rotations de chaque articulation, nous calculons d'abord la différence des positions de cette articulation et de son articulation parente, ce qui nous permettra d'obtenir la direction du lien parente, puis nous calculons la rotation de ce lien qui va de l'ancienne à la nouvelle direction en utilisant l'équation suivante :

$$\theta = (\overrightarrow{CM}, \overrightarrow{CM'}) \quad (3.16)$$

Tel que :

C : est le point centre de la rotation ou l'os parent de l'articulation.

M : l'ancienne position de l'articulation avant d'avoir effectué un mouvement.

M' : la nouvelle position de l'articulation après avoir effectué un mouvement.

θ : l'angle de rotation de l'articulation.

Nous affecterons alors l'angle de cette rotation à l'articulation, un exemple de cette opération est expliqué dans la figure 3.22:

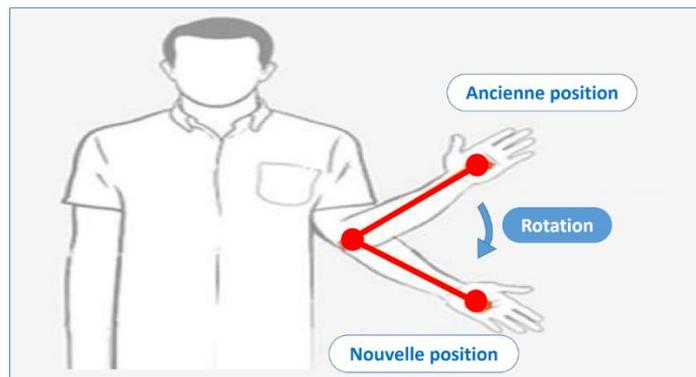


Figure 3.22 : Exemple du calcul de la rotation de l'articulation de la main gauche

5.3.2.2 Adaptation du modèle 3D de l'Avatar avec la nature du modèle squelettique

Dans le but de reproduire avec fiabilité les mouvements exercés par le sujet humain dans la scène réelle, nous avons besoin d'un modèle 3D d'un Avatar qui représentera l'homologue du sujet humain dans la scène virtuelle.

- **Modèle 3D de l'Avatar**

Le choix du modèle 3D de l'Avatar doit être en accord avec la nature du modèle des données envoyées par le capteur d'acquisition (données squelettiques). Ces modèles 3D considèrent le corps humain comme étant une chaîne cinématique représentée sous forme d'arbre hiérarchique. Il forme un ensemble de membres reliés par des articulations qui constituent le squelette du modèle 3D (voir figure 3.23).



Figure 3.23 : Modèle d'Avatar

Le mouvement humain est décrit sous la forme de rotation et de translation de ces articulations qui constituent le modèle 3D de l'Avatar. Ce dernier varie en forme et en style, ainsi qu'en nombre d'os et d'articulations selon les détails attendus et le type du système d'acquisition.

Comme notre modèle squelettique fourni par le capteur Kinect est constitué de vingt articulations. Pour cela, nous utilisons dans notre cas un modèle d'Avatar 3D de forme humaine constitué du même nombre d'articulations afin de simuler les mouvements réels du sujet humain d'une manière appropriée et cohérente dans le monde virtuel et en temps réel. Nous faisons correspondre chaque articulation du modèle des données squelettiques fournies par le module fusion à son articulation homologue du squelette du modèle de l'Avatar, comme l'illustre la figure 3.24.

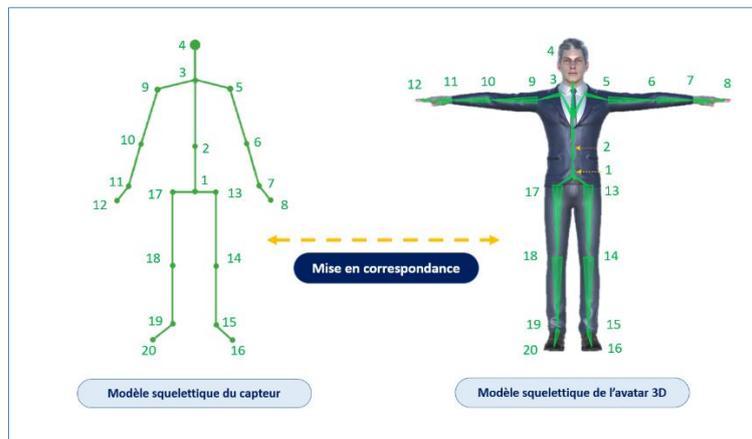


Figure 3.24 : Mise en correspondance entre le squelette de la Kinect et le squelette de l'Avatar

5.3.2.3 L'animation de l'Avatar

Le processus de l'animation de l'Avatar dans un monde virtuel est la dernière étape effectuée sur les données squelettiques fournies par les différents modules expliqués précédemment. Il consiste en premier lieu à réceptionner le modèle squelettique fusionné résultant du module de la fusion pour l'adapter par la suite au squelette interne de l'Avatar présélectionné.

Dans le but de garder la continuité dans le mouvement de l'animation, le modèle de l'Avatar est mis à jour à chaque frame par les données squelettiques reçues, et ceci en modifiant la rotation de chaque articulation de l'Avatar par celle du squelette fusionné. Ainsi, c'est de cette façon que nous déplaçons et animons notre Avatar en reproduisant les mêmes mouvements qu'effectue notre sujet humain en temps réel. Si le sujet humain disparaît du champ de vision

des capteurs, nous remettons l'Avatar à sa position initiale avec les rotations d'articulations initiales.

Ce processus permet au sujet humain de se déplacer librement et sans aucune contrainte sur un champ de 360° et nous assure une animation lisse, fluide et réaliste représentant l'exactitude des mouvements du monde réel dans le monde virtuel.

5.4 Sauvegarde et représentation graphique de données squelettiques

Afin d'analyser et d'exploiter les données squelettiques fusionnées et de fournir la fonctionnalité de l'animation différée, nous avons conçu une méthode permettant de sauvegarder les données squelettiques résultantes des mouvements du sujet humain.

Nous avons alors élaboré une structure de données simple et efficace pour stocker les données squelettiques. Nous avons utilisé la notation JSON qui est un format de fichier open-standard permettant de stocker les données de manière organisée et lisible tout en y facilitant l'accès. Les données sont présentées dans un format textuel constitué de paires « clé / valeur ». Ce format peut toutefois être généré et lu par la plupart des langages de programmation, ce qui facilite les futures extensions de notre système. La structure de notre fichier JSON de sauvegarde de l'animation est illustré sur la figure 3.25. Pour chaque animation, nous créons un fichier nommé par l'identifiant de l'animation contenant une liste des frames. Chaque frame est constitué de vingt articulations qui sont défini par leurs positions et rotations.

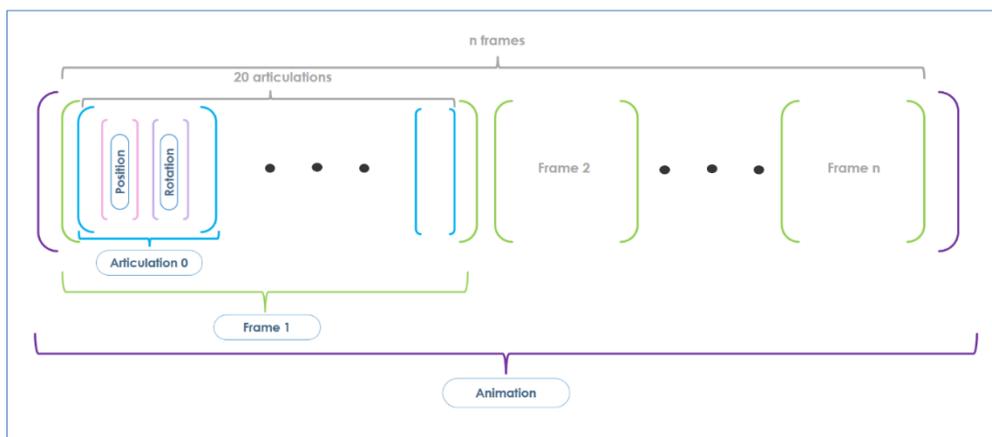


Figure 3. 25 : Structure JSON des données d'une animation

La figure 3.26 montre un exemple d'une partie des données sauvegardée d'un frame d'une animation :

```

1 {"AnimationData": {
2   "articuPosition":
3     [
4       {"x": 0.000026696947315940633, "y": 0.9997748136520386, "z": 0.00454754557363987},
5       {"x": 0.0, "y": 0.09694427251815796, "z": -0.011097324080765248},
6       {"x": 0.0, "y": 0.1463589586734773, "z": -2.8285999320587508e-8},
7       {"x": 0.0, "y": 0.06944289803504944, "z": 0.019824808463454248},
8       {"x": -0.06681303679943085, "y": 0.12887641787528993, "z": -0.0015732940519228578},
9       {"x": -4.044393292490511e-11, "y": 0.1401614397764206, "z": 4.1285630469900527e-10},
10      {"x": -2.2562728074770889e-10, "y": 0.2733026146888733, "z": -5.332679783864869e-11},
11      {"x": 9.546457374609574e-11, "y": 0.2088880836963537, "z": 3.664894221433457e-10},
12      {"x": 0.035568881779909137, "y": 0.11070998013019562, "z": 0.00153639807831456},
13      {"x": 0.06681303679943085, "y": 0.12897330522573232, "z": -0.0024197502061724664},
14      {"x": 9.148960755656077e-11, "y": 0.1401614397764206, "z": 4.067513270644696e-10},
15      {"x": -3.004787807459053e-10, "y": 0.2731824517250061, "z": -3.198641629342802e-10},
16      {"x": 8.802657908146827e-11, "y": 0.20889824628829957, "z": 5.525663260286251e-11},
17      {"x": -0.03650405630469322, "y": 0.1121010293460846, "z": 0.0008198461728170514},
18      {"x": -0.08726321905851364, "y": -0.05393180623650551, "z": -0.0044503449292183},
19      {"x": -1.0183850429612563e-9, "y": 0.4105736017227173, "z": -7.2578183552764e-11},
20      {"x": -5.66295765927032e-10, "y": 0.42036898933124542, "z": -5.308817274674027e-10},
21      {"x": -3.4968272718692e-10, "y": 0.1868141293526959, "z": -7.826384873510506e-9},
22      {"x": 0.08726321905851364, "y": -0.05393180623650551, "z": -0.00458878443935871},
23      {"x": 2.507781760030525e-10, "y": -0.4392683506011963, "z": 0.5642642974853516, "w": 0.42960256338119509},
24      {"x": 2.2851821823710595e-10, "y": 0.41974708437919619, "z": 1.199959791742249e-10},
25      {"x": 8.24308282941621e-10, "y": 0.1821046769618988, "z": 5.342026265786615e-10}
26    ]
27   "articuRotation":
28     [
29       {"x": 0.047406695783138278, "y": 0.07072773575782776, "z": 0.03208182007074356, "w": 0.995851874315015},
30       {"x": -0.09850010275840759, "y": -0.017036426812410356, "z": 0.009512652643024922, "w": 0.994945764541626},
31       {"x": 0.09572658687829972, "y": 0.006260029040277004, "z": 0.01426449034969807, "w": 0.995285809400696},
32       {"x": 0.0060876822095908962, "y": 0.00036105187609791757, "z": -0.011666317470371724, "w": 0.999609571495056},
33       {"x": 0.5514443516731262, "y": -0.4392683506011963, "z": 0.5642642974853516, "w": 0.42960256338119509},
34       {"x": 0.371289998292923, "y": 0.017129985615611078, "z": 0.05023045837879181, "w": 0.926999021020508},
35       {"x": -0.08845486491918564, "y": -0.2244851291179657, "z": -0.2561459541320801, "w": 0.9360402822494507}
36     ]
37 }
    
```

Figure 3.26 : Exemple de sauvegarde des données squelettiques d'une animation

Les données de chaque animation sont aussi sauvegardées sous forme d'un fichier CSV qui est un format de texte simple sous forme d'un tableau. Chaque ligne du fichier correspond à une ligne du tableau « Frames ». Les valeurs de chaque colonne du tableau « articulations » sont séparées par un caractère de séparation « Point-Virgule ». Le but principal de ce fichier dans notre cas est de permettre au médecin traitant d'exploiter les données squelettiques sauvegardées par d'autres logiciels tels que Microsoft Excel ou autres pour lui fournir les différentes représentations graphiques de ces données (Voir la figure 3.27).

The image shows a CSV file and its corresponding Excel spreadsheet. The CSV file contains a list of frames with joint coordinates (x, y, z, w) for various body parts like HipCenter, Spine, Shoulder, Head, etc. The Excel spreadsheet shows the same data in a grid format with columns labeled A through U and rows labeled Frame 1 through Frame 18. The CSV file is labeled 'CSV' and the Excel spreadsheet is labeled 'Xls'.

Figure 3.27 : Exemple de fichier sauvegarder sous-format CSV exploitable par Excel

Au niveau de notre système, nous avons conçu un outil de représentation 3D des données squelettiques enregistrées permettant au médecin traitant de visualiser ces données en sélectionnant les articulations et les os pour observer leurs mouvements. Cela permet de voir les mouvements des différentes parties du corps du sujet humain d'une façon séparée. Ainsi que d'analyser la variation des positions de chaque articulation en fonction du temps. Un exemple de représentation 3D de la partie inférieure du corps est donné dans la figure 3.28.

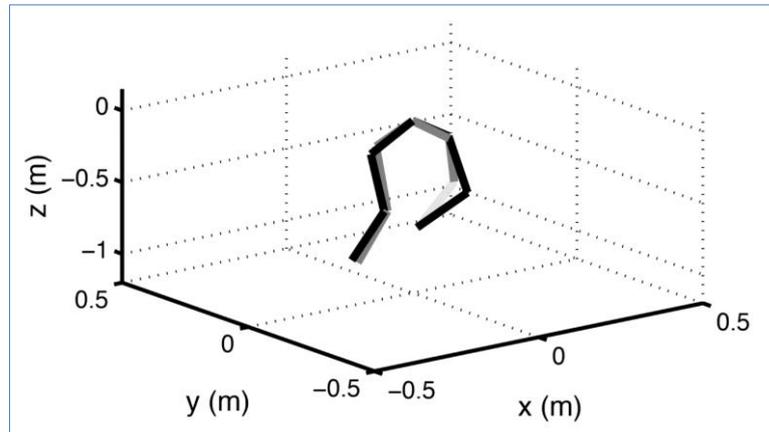


Figure 3.28 : Exemple de représentation 3D de la partie inférieure du corps

Cet outil permet aussi de visualiser les différentes postures réalisées par le sujet humain dans chaque frame permettant au médecin traitant d'examiner ces postures et d'identifier les maladies et les anomalies de la marche du sujet humain.

5.5 L'animation différée

Ce module permet au médecin traitant de reproduire les mouvements exercés par le patient (le sujet humain dans notre cas), lors d'une séance de rééducation en mode différé pour diagnostiquer d'éventuelles maladies ou anomalies de sa marche, par un Avatar dans un environnement virtuel. Le médecin traitant peut aussi comparer les différents mouvements réalisés par le même patient durant plusieurs séances de rééducation.

Pour l'animation différée, comme le montre la figure 3.29, nous identifions en premier lieu le fichier JSON de l'animation sélectionnée par le médecin traitant afin de récupérer les données squelettiques représentatives des mouvements réels exercés par le sujet humain lors de la séance de l'examen de rééducation. Puis nous réadaptions ces données au fur et à mesure à l'Avatar dans la scène virtuelle.

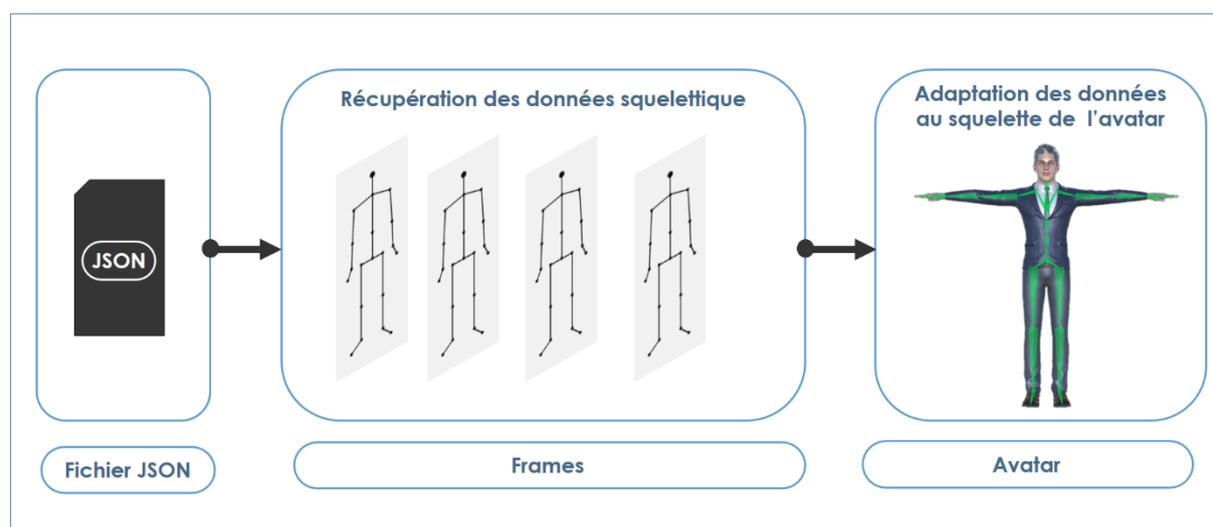


Figure 3.29 : Processus d'animation différée

6 Conclusion

Dans ce chapitre nous avons présenté les étapes de la conception de notre système de capture de mouvement multi-Kinect pour l'animation d'Avatars. Nous avons commencé par décrire l'architecture globale de notre système où nous avons décrit le fonctionnement des principaux modules qui la constitue. Nous sommes passés par la suite à modéliser le système proposé en utilisant la notation UML via ces différents diagrammes. Puis nous avons détaillé les différents modules, en commençant par le module d'acquisition et de prétraitement, passant par le module de calibrage pour finir avec le module d'animation par la fusion des données. Nous avons aussi présenté le processus de sauvegarde des données squelettiques fusionnées et leur lecture pour l'animation différée de l'Avatar. Ainsi que le processus de représentation graphique de ces données afin de permettre aux médecins traitants de les analyser pour détecter les maladies et les anomalies dans la marche du patient.

Dans le chapitre suivant, nous présenterons l'implémentation de notre système, ainsi que des discussions sur les résultats et les tests que nous avons obtenus lors de l'évaluation de notre système.



Chapitre 4

Réalisation et tests



1 Introduction

Nous consacrons l'intégralité du présent chapitre au développement des différents modules qui constituent notre système afin d'aboutir à une implémentation concrète de notre approche. Nous présentons en premier lieu, les différentes technologies et outils utilisés pour implémenter notre solution en évoquant le matériel, les langages de programmation et les outils informatiques utilisés. Puis, nous présentons le produit résultant de notre travail en détaillant son fonctionnement, ses interfaces et mode d'utilisation. Nous clôturons ce chapitre par une série d'expérimentations réelles et de tests effectués avec notre système passant par la configuration réelle des capteurs ainsi que l'évaluation des performances temporelles de notre système.

2 Technologies et outils utilisés

Dans le cadre de la mise en œuvre de notre système, nous avons utilisé différentes technologies et outils informatiques existants, testés par la communauté scientifique. Ils nous ont permis de résoudre les problèmes que nous avons rencontré pour élaborer notre application. Nous avons fait usage des environnements suivants :

2.1 Environnement matériel

Nous avons utilisé un environnement matériel constitué de cinq ordinateurs clients, de cinq capteurs Kinect branchés aux machines clientes et d'un ordinateur serveur possédant les caractéristiques illustrées dans le tableau 4.1. En plus, nous avons utilisé quelques équipements informatiques qui font office de réseau local, reliant les différentes machines utilisées (les clients et le serveur). L'ensemble du matériel informatique utilisé est dressé dans les tableaux 4.1 et 4.2.

Tableau 4.1 : Caractéristiques des machines utilisées

	Marque	Processeur	Ram	Système d'exploitation	Carte Graphique
PC 1	PC portable DELL	Intel® Core(TM) i7- 8550U CPU @1.80GHz	8Go	Windows 10 professionnel 64 bit	Intel UHD Graphics 620 4 GO
PC 2	PC portable FUJITSU	Intel® Core(TM) i7- 7500 CPU @2.70GHz	8Go	Windows 10 professionnel 64 bit	Intel UHD Graphics 620 4 GO

PC 3	PC portable FUJITSU	Intel® Core(TM) i7- 7500 CPU @2.70GHz	8Go	Windows 10 professionnel 64 bit	Intel UHD Graphics 620 4 GO
PC 4	PC portable Lenovo	Intel® Core(TM) i3- 6100 CPU @2.30GHz	4Go	Windows 10 professionnel 64 bit	Intel(R) HD Graphics 520
PC 5	PC de bureau HP Pro	Intel® Core(TM) i5- 3470 @3.60GHz	4Go	Windows 7 professionnel 64 bits	Intel(R) HD Graphics 2500
PC 6	PC de bureau Dell	Intel Core i5- 3340U CPU @3.10GHz	8GO.	Windows 7 professionnel 64 bits	Intel(R) HD Graphics 2500

Tableau 4.2 : Matériel informatique utilisé

Un Switch Ethernet
Six Câbles Ethernet
Adaptateurs USB des capteurs Kinect
5 Kinect XBOX 360

2.2 L'environnement logiciel

Pour la mise en œuvre des différentes étapes discutées précédemment, nous avons opté pour des outils largement utilisés par la communauté scientifique, des technologies reconnues fiables et des versions stables afin d'assurer la fluidité de la mise en œuvre de notre solution et d'éviter ainsi les problèmes de compatibilité.

2.2.1 Langage C#

Le C# est un langage de programmation orienté objet, fortement typé, dérivé de C et de C++. Il fournit une prise en charge complète de la programmation orientée objet, notamment l'abstraction, l'encapsulation, l'héritage et le polymorphisme. Le langage C# est donc considéré comme un langage de haut niveau et destiné à développer sur la plateforme .NET où la version utilisée dans notre système est '2.0'.



Figure 4.1 : Logo C#

2.2.2 JSON

La notation objet du langage *Javascript* JSON est un format d'échange de données en texte lisible. Il est utilisé pour représenter et stocker des structures de données et des objets simples de manière organisée. Il a l'avantage d'être facilement lisible par un être humain et permet un accès rapide aux données qu'il contient. Bien que JSON puise sa syntaxe du *JavaScript*, il est indépendant de tout langage de programmation. Il peut ainsi être interprété par tout langage à l'aide d'un parseur.

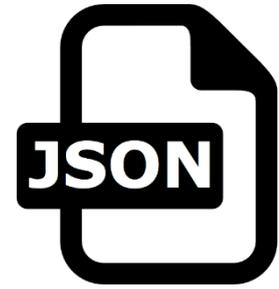


Figure 4.2 : Logo JSON

2.2.3 CSV

Comma-separated values, connu sous le sigle CSV, est un format texte ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules. Ce format permet de traiter et de stocker de grosses quantités de données structurées dans un fichier exploitable par divers autres logiciels tel que Excel(Microsoft) et Calc (*LibreOffice*), ou importable par les bases de données comme MySQL et Oracle.



Figure 4.3 : Logo CSV

2.2.4 Microsoft Kinect SDK 1.8

Kit de développement officiel de la Kinect Xbox 360 pour Windows est un ensemble de bibliothèques développées par Microsoft, permettant un accès facile aux différentes fonctionnalités du périphérique Kinect.



Figure 4.4 : Logo Microsoft Kinect SDK

2.2.5 Microsoft Visual Studio 2017

Outil de développement représentant une suite de logiciels de développement pour Windows conçu par Microsoft, qui utilisent l'environnement de développement intégrée (IDE).



Figure 4.5 : Logo Visual Studio

2.2.6 Unity 5.6.0f3

Moteur de jeux open source, développée par *Unity Technologies*, permet de créer un contenu réellement interactif, d'importer des modèles 3D, de les contrôler et de gérer leurs comportements dans l'environnement virtuel grâce à des scripts écrits en C# ou en JavaScript.



Figure 4.6 : Logo Unity

2.2.7 Mixamo

C'est une application web open source, développée par la communauté Adobe. Elle permet de créer et de générer des modèles humains 3D virtuel et des animations. Nous l'avons utilisé dans le but de créer des modèles 3D (Avatars) articulés qui conviennent à notre application.



Figure 4.7 : Logo Mixamo

2.2.8 Autodesk 3DS Max 2016

C'est un logiciel développé par la communauté Autodesk aide à la modélisation et l'animation 3D. Il permet de créer des modèles 3D, de les texturer et de les exporter sous plusieurs formats de fichiers.



Figure 4.8 : Logo AUTO DESK

3 Présentation du système développé

Nous allons présenter dans cette section le système que nous avons développé, ses différents modules ainsi que son principe de fonctionnement.

3.1 Authentification et inscription

Nous commençons tout d'abord par exposer une interface d'authentification qui apparaît au lancement de notre application (voir figure 4.9). Cette dernière est obtenue après avoir invoqué notre application.

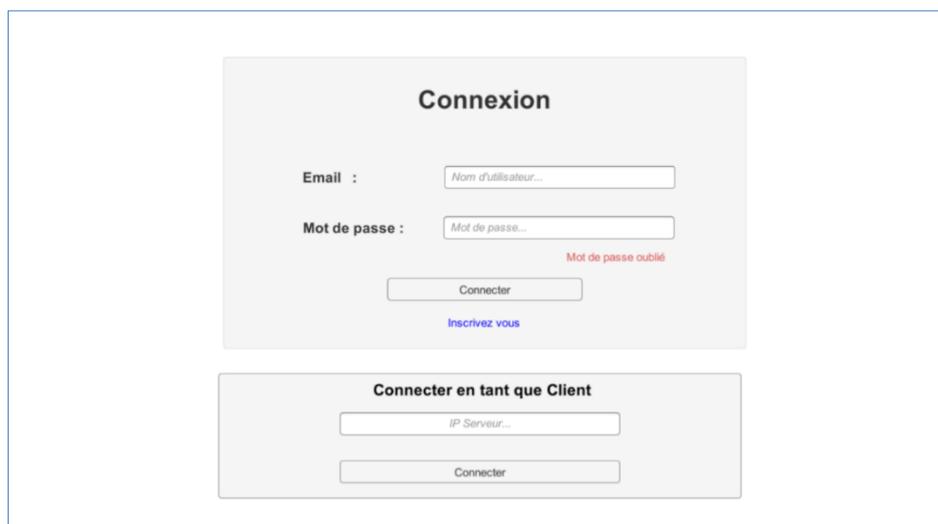


Figure 4.9 : Interface d'authentification

Si l'utilisateur a lancé une machine client, il accède directement à la scène de la capture de mouvements en spécifiant l'adresse IP du serveur et en cliquant sur le bouton « Connecter ».

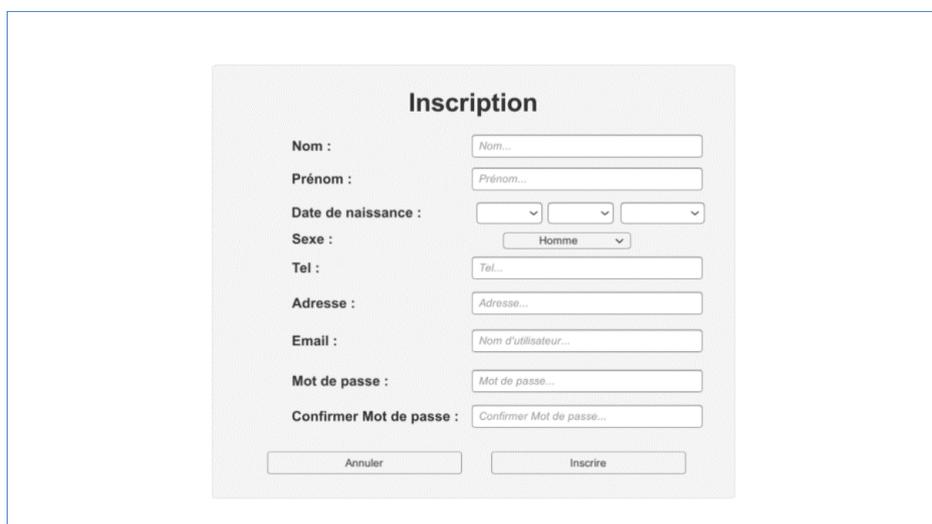


Figure 4.10 : Scène de capture de mouvements « Client »

La scène de capture client illustrée dans la figure 4.10 est utilisée pour l'animation d'un avatar par les données locales récoltées par le capteur. Elle permet aussi de visualiser les images RGB acquises par le capteur. La barre de défilement qui se trouve à gauche de l'écran permet d'incliner le capteur en haut et en bas afin de bien viser le champ de vision du capteur.

Cependant, le lancement de la machine serveur se fait par authentification en introduisant un email et un mot de passe valides, afin de pouvoir accéder au menu principal de l'application du

système. L'utilisateur peut aussi s'inscrire à l'aide de l'interface illustré dans la figure 4.11 en cliquant sur le bouton « inscrivez-vous » (figure 4.9).

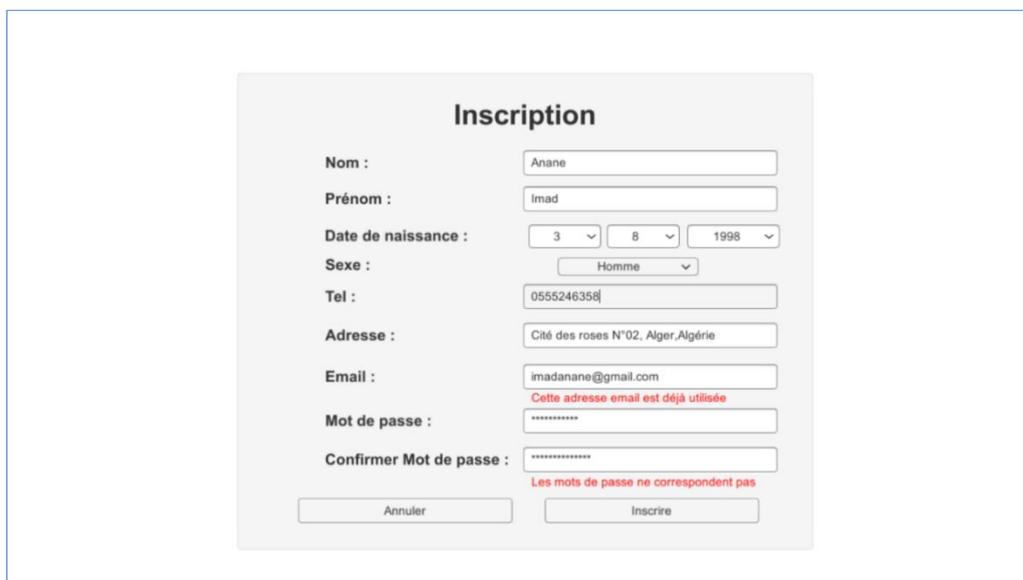


The image shows a registration form titled "Inscription". It contains the following fields and controls:

- Nom :
- Prénom :
- Date de naissance : Three dropdown menus for day, month, and year.
- Sexe : A dropdown menu with "Homme" selected.
- Tel :
- Adresse :
- Email :
- Mot de passe :
- Confirmer Mot de passe :
- Buttons: "Annuler" and "Inscrire".

Figure 4.11 : Interface d'inscription

Après le remplissage du formulaire d'inscription et en cliquant sur le bouton « inscrire », le système élabore une étape de vérification des données saisies. Si les données ne correspondent pas aux données du personnel de la structure, des messages d'erreur sont affichés comme illustre la figure 4.12.



The image shows the same registration form as in Figure 4.11, but with the following data entered and error messages displayed:

- Nom :
- Prénom :
- Date de naissance : 3 / 8 / 1998
- Sexe : Homme
- Tel :
- Adresse :
- Email :
Cette adresse email est déjà utilisée
- Mot de passe :
- Confirmer Mot de passe :
Les mots de passe ne correspondent pas
- Buttons: "Annuler" and "Inscrire".

Figure 4.12 : Interface d'inscription après vérification des données saisies

Si les données introduites correspondent bien aux données de l'un des médecins traitants du service, un email contenant un code de confirmation est envoyé à l'adresse email spécifiée comme montre la figure 4.13.

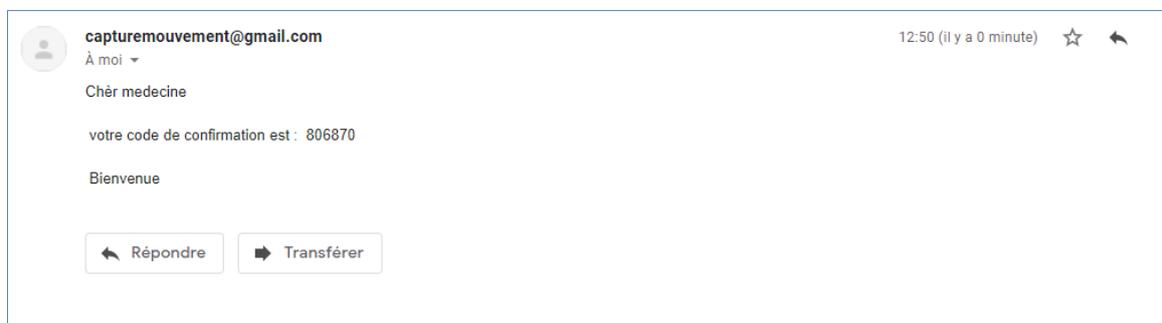


Figure 4.13 : Email de confirmation

L'utilisateur saisie le code de confirmation qu'il a reçu dans le champ « code » de l'interface, puis il clique sur le bouton confirmer comme l'illustre la figure 4.14. Si le code saisi est incorrect un message d'erreur est affiché sinon le menu principal de l'application du système est affiché.

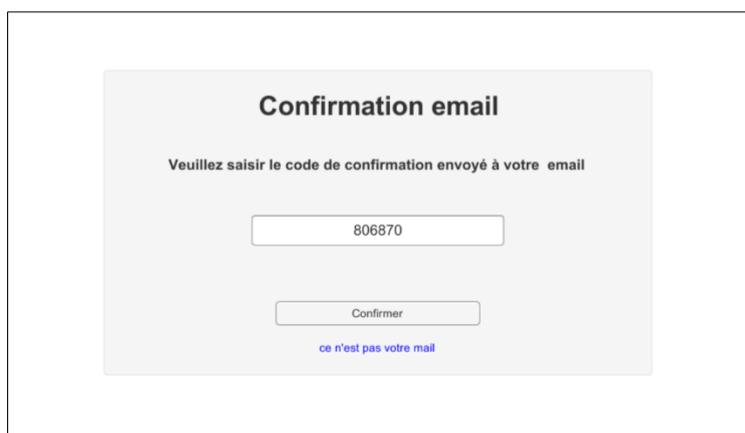


Figure 4.14 : Interface de confirmation d'authentification

Après cette étape, l'utilisateur est authentifié par le système comme étant personnel traitant appartenant à la structure de santé dans laquelle le système est installé et peut accéder aisément au système.

Au cas où l'utilisateur oublie son mot de passe, il peut le changer en cliquant sur le bouton « mot de passe oublié » de l'interface d'authentification (voir figure 4.9). Le système demandera alors à l'utilisateur de saisir son email en premier lieu via l'interface de récupération d'email (figure 4.15).



Figure 4.15 : Interface de récupération de l'email pour le changement du mot de passe

Puis, pour confirmer la propriété du compte à l'utilisateur, un email contenant un code de confirmation est envoyé à l'adresse spécifiée. Ce dernier sera saisi par l'utilisateur dans l'interface de la figure 4.16.

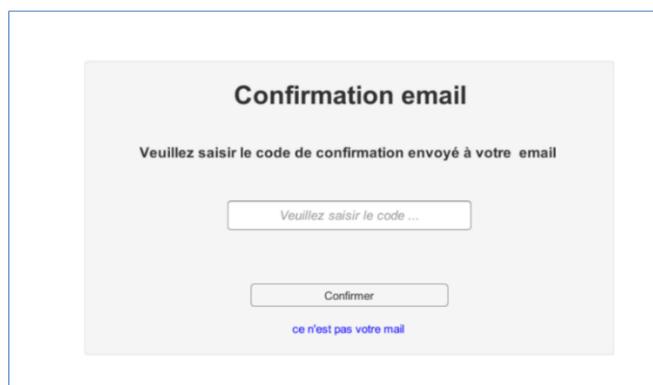


Figure 4.16 : Interface du changement du mot de passe

Après la saisie du code et la confirmation, une interface de changement du mot de passe est affichée (Figure 4.17) celle-ci permet à l'utilisateur de renouveler son mot de passe.



Figure 4.17 : Interface du changement du mot de passe

3.2 Menu principal

Après l'authentification de l'utilisateur, le menu principal sur la machine serveur est affiché comme montre la figure 4.18.

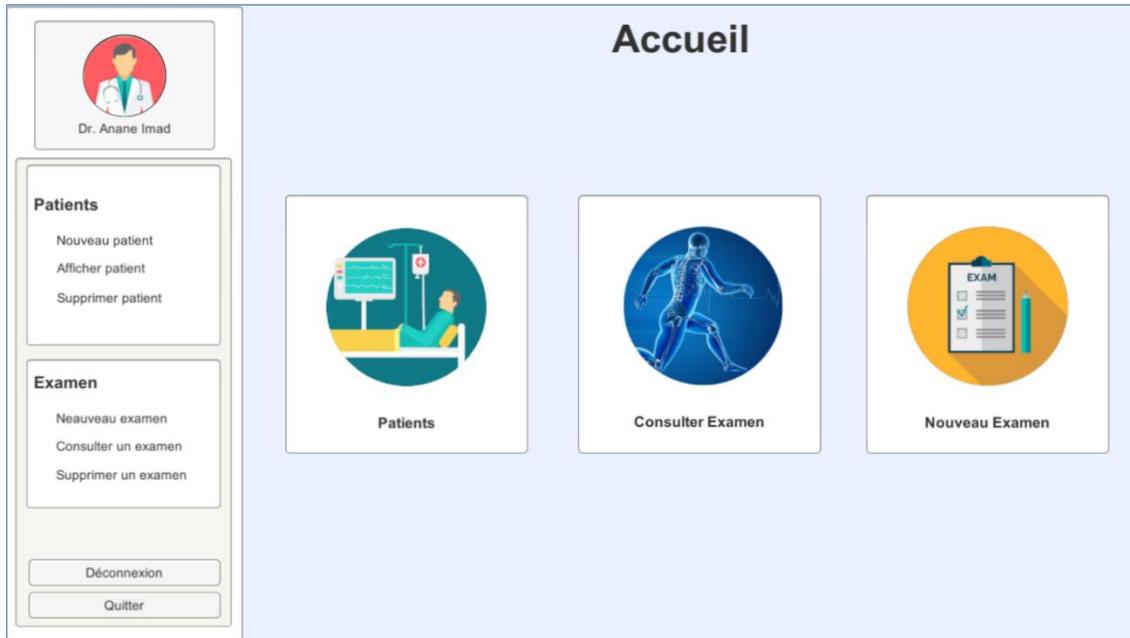


Figure 4.18 : Interface du menu principal sur la machine serveur

Le menu principal de la machine serveur est constitué de plusieurs panneaux, chacun d'eux fournit une fonctionnalité bien déterminée du système :

- Le panneau du profil « Médecin »
- Le panneau « Gestion des patients »
- Le panneau « Nouveau examen »
- Le panneau « Consulter examen »

3.2.1 Le panneau du profil « Médecin »

Le panneau « Profil » de la figure 4.19 est consacré à l'affichage des informations propres au médecin. Il permet aussi de les modifier en cliquant sur ce panneau via l'interface de modification du profil (voir figure 4.20). L'utilisateur "Médecin" a été utilisé dans notre cas à titre d'exemple car ce dernier pourrait avoir besoin de ce type de plateforme pour

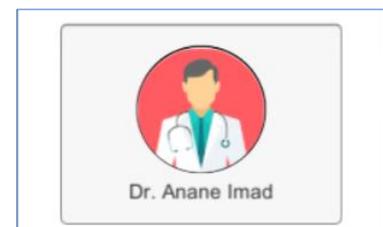


Figure 4.19 : Panneau Profil

améliorer son diagnostic dans le cadre d'une rééducation fonctionnelle d'un membre donné.

The screenshot shows a web interface for a doctor's profile. On the left, a sidebar contains a profile picture of Dr. Anane Imad and two main sections: 'Patients' with options 'Nouveau patient', 'Afficher patient', and 'Supprimer patient'; and 'Examen' with options 'Neauveau examen', 'Consulter un examen', and 'Supprimer un examen'. At the bottom of the sidebar are 'Déconnexion' and 'Quitter' buttons. The main area is titled 'Profil' and contains a form with the following fields: 'Nom' (Anane), 'Prénom' (Imad), 'Date de naissance' (3/8/1998), 'Sexe' (Homme), 'Tel' (0555246358), 'Adresse' (Cité les roses N°02, Alger, Algérie), 'Mot de passe' (Mot de passe...), and 'Confirmer Mot de passe' (Confirmer Mot de passe...). A 'Modifier Email' button is centered below the form, and 'Annuler' and 'Enregistrer' buttons are at the bottom.

Figure 4.20 : Interface de modification du profil

Le bouton « Quitter » permet de fermer la session courante et de quitter l'application tandis que le bouton « Déconnexion » ferme juste la session et affiche à nouveau l'interface d'authentification.



Figure 4. 21 : Bouton « Déconnexion » et « Quitter »

3.2.2 Le panneau « Gestion des patients »

Le panneau Patients illustré dans la figure 4.22 permet d'afficher l'interface de gestion des patients (voir figure 4.23)

Cette interface permet au médecin traitant de voir la liste de ses patients dans le système via un tableau illustrant leurs informations. Ainsi que d'ajouter des nouveaux patients, d'afficher, modifier ou supprimer les patients guéris.

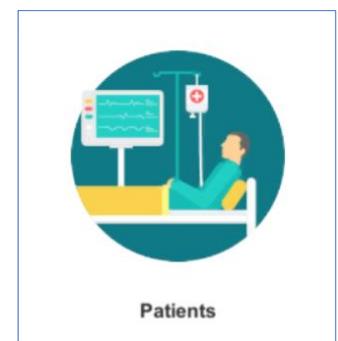


Figure 4.22 : Panneau Patients



Figure 4.23 : Interface gestion des patients

Les boutons « Afficher », « Modifier » et « Supprimer » sont désactivés jusqu'à ce que l'utilisateur sélectionne un patient du tableau des patients.

En cliquant sur le bouton « ajouter », l'interface de la figure 4.24 est affichée dans laquelle l'utilisateur spécifie les informations du patient à ajouter.



Figure 4.24 : Interface ajouter patient

Le bouton « Modifier » permet d'afficher l'interface de modification des informations du patient sélectionné du tableau (figure 4.25). Tandis que le bouton « Supprimer » permet de supprimer le patient sélectionné du tableau.



Figure 4.25 : Interface modifier patient

Le bouton « Afficher » permet d’afficher toutes les informations du patient sélectionné ainsi que tous ses examens effectués au sein de la structure sanitaire en spécifiant la date et le médecin superviseur de chaque examen (voir la figure 4.26). Cette interface contient aussi un bouton raccourci pour la consultation ou la suppression d’un examen de la liste.

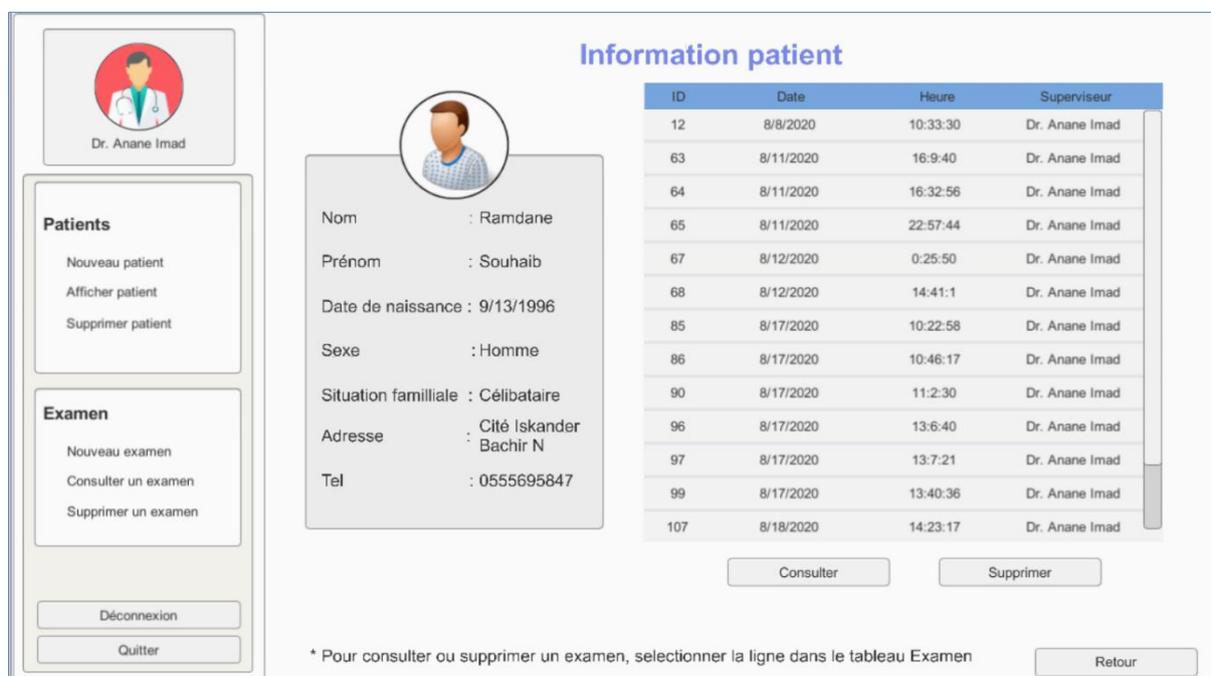


Figure 4.26 : Interface afficher informations patient

3.2.3 Le panneau « Nouveau Examen »

Le panneau « Nouveau examen » du menu principal illustré, dans la figure 4.27, permet d’afficher l’interface de sélection du patient souhaitant lui faire un examen (voir figure 4.28).



Figure 4.27 :Panneau nouveau examen

The screenshot shows a web interface titled 'Selection du patient'. At the top left, there is a profile for 'Dr. Anane Imad' with a doctor icon. Below this are two menu sections: 'Patients' with options 'Nouveau patient', 'Afficher patient', and 'Supprimer patient'; and 'Examen' with options 'Nouveau examen', 'Consulter un examen', and 'Supprimer un examen'. At the bottom left are buttons for 'Déconnexion' and 'Quitter'. The main area contains a table of patients with the following data:

ID	Nom	Prenom	Date nais	Sexe	Situation	Tel
13	Djekoune	Walid	3/6/1981	Homme	Marié(e)	0777586932
14	Ramdane	Souhaib	13/9/1996	Homme	Célibataire	0555695847
16	Zola	Leila	12/9/1976	Femme	Marié(e)	0791253614
17	Hadjersi	Mohammed	11/7/1996	Homme	Célibataire	0668241532
18	Gizelaine	Sara	19/10/1991	Femme	Célibataire	0569878421
19	Kalila	Anis	5/9/2005	Homme	Célibataire	0663214596
20	Ousalah	Ihcen	8/12/1994	Femme	Célibataire	0669632001
21	Osmane	Malika	19/1/1991	Femme	Marié(e)	0687695874

At the bottom of the interface are two buttons: 'Continuer' and 'Retour'.

Figure 4.28: Interface sélection du patient pour un examen

Après la sélection du patient et en cliquant sur le bouton « Continuer », le serveur est lancé et l’interface de connexion des capteurs Kinect est affichée en demandant au médecin/utilisateur de connecter toutes les machines Clients à l’adresse IP du serveur spécifiée en bas de l’écran comme le montre la figure 4.29.

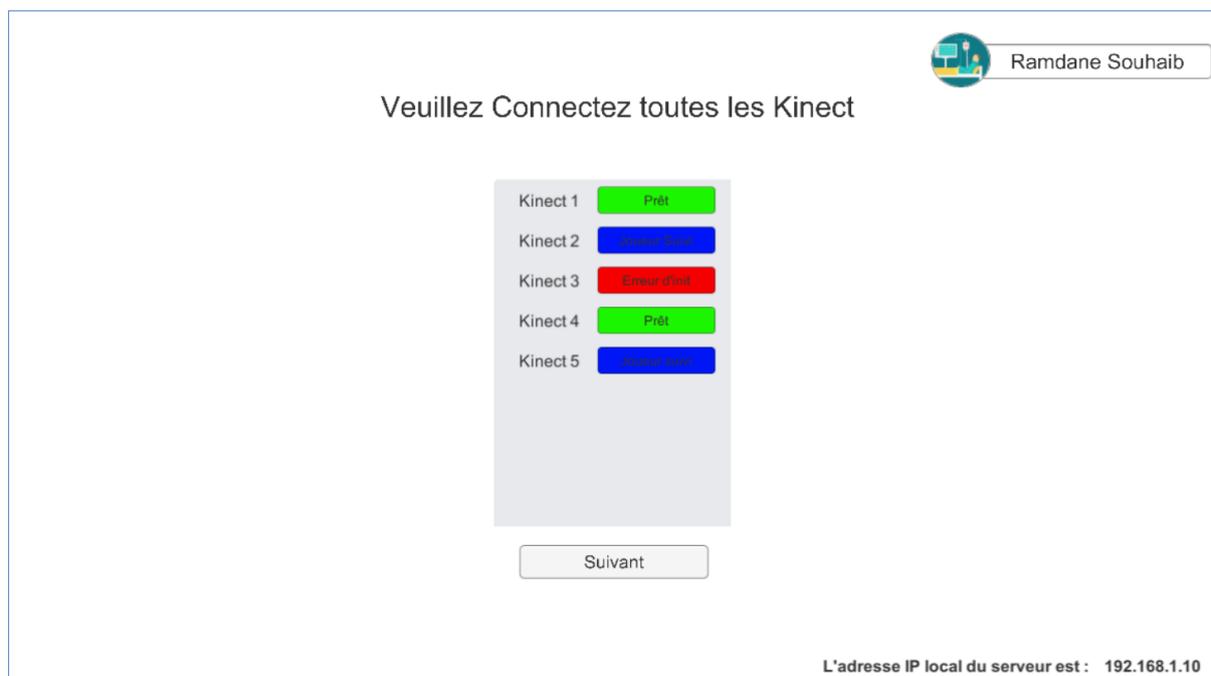


Figure 4.29 : Interface de connexion des capteurs Kinect

L'interface contient des informations concernant l'état de suivi du patient par chaque capteur Kinect client du réseau ainsi que sa configuration dans la scène :

- Le nom affecté à chaque client.
- L'état d'initialisation du capteur Kinect qui peut prendre les formes suivantes : Capteur Kinect Prêt (bien initialisé) / Erreur d'initialisation / Sujet humain suivi par le capteur.

L'état du capteur Kinect peut prendre trois couleurs différentes, à savoir :

- Rouge : désigne qu'il y a eu une erreur d'initialisation du capteur Kinect.
- Vert : désigne que l'initialisation a été effectuée avec succès et que le capteur Kinect est prêt pour la capture.
- Bleu : désigne que le squelette du patient est bien suivi par le capteur.

Après que l'utilisateur connecte tous les capteurs Kinect au serveur, l'application lui demande de guider le patient à se mettre entre chaque combinaison des Kinect, le système calcule les matrices de transformation « Processus de calibrage » comme le montre la figure 4.30. Cette étape peut se faire une seule fois au début après que l'ensemble des capteurs Kinect sont installés, en supposant que ces capteurs sont fixes et ne bougent pas.

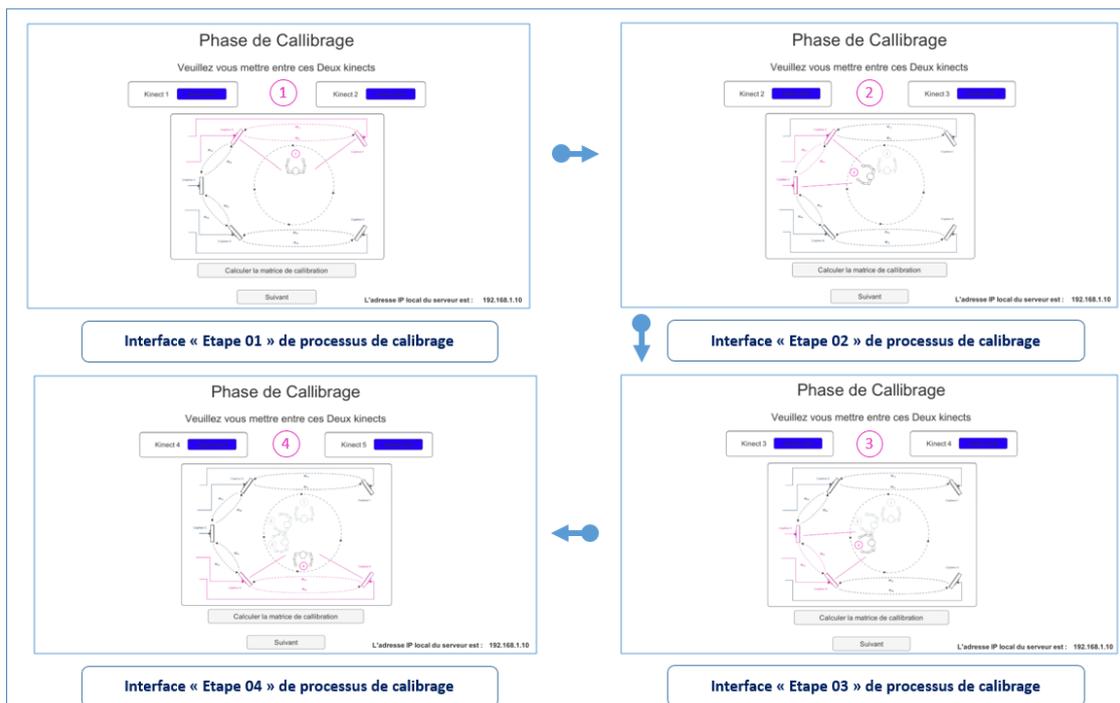


Figure 4.30 : Interfaces des étapes du processus de calibration des capteurs

Dès que le processus de calibration des capteurs est réalisé avec succès, le système illustre directement la scène de l'animation 3D de l'avatar fusionné ainsi que les avatars reconstruits par les machines clients. La scène contient toutes les informations concernant le patient examiné, l'état de suivi de chaque Kinect ainsi qu'un ensemble de bouton fournissant diverses fonctionnalités (voir la figure 4.31).



Figure 4.31 : Scène d'animation 3D en temps réel

- Le bouton « Calibrer » permet de refaire le processus de calibrage au cas où il y a eu un déplacement de l'un des capteurs Kinect du réseau utilisé ou un changement dans la configuration matérielle initiale du système.
- Le bouton « Retour » permet de retourner à l'interface de sélection du patient pour effectuer un nouvel examen.
- Le bouton « Enregistrer » permet de sauvegarder les données squelettiques des mouvements virtuels animés par l'avatar pour une utilisation ultérieure. Après la fin de l'enregistrement le médecin/utilisateur pourra rajouter des notes et des remarques sur l'examen effectué via l'interface d'enregistrement des notes (figure 4.32).

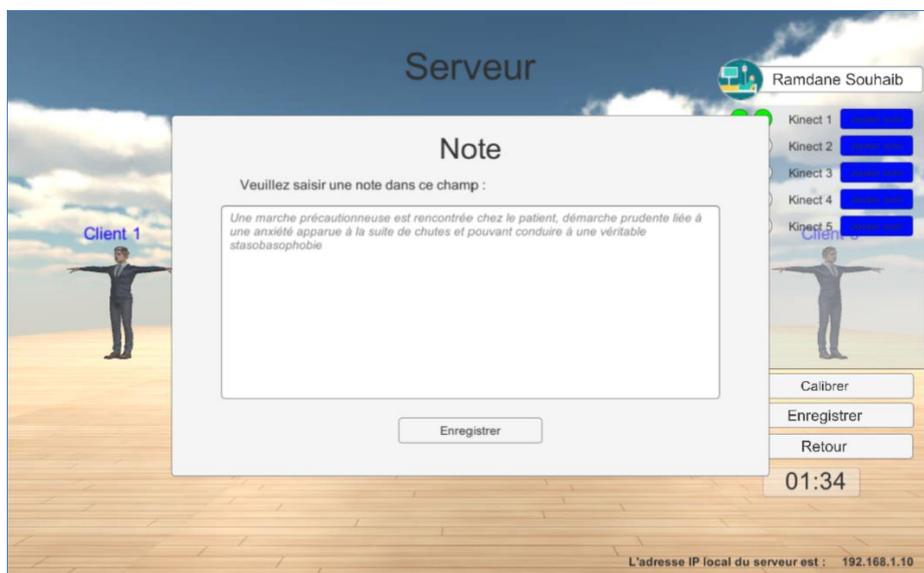


Figure 4.32 : Fenêtre d'enregistrement des notes

La liste illustrée dans la figure 4.33 fournit en temps réel des informations pertinentes concernant les états du suivi des capteurs (la zone 3), l'état de détection de visage (la zone 2) ainsi que le repère de base utilisé dans la fusion des données (la zone 1).

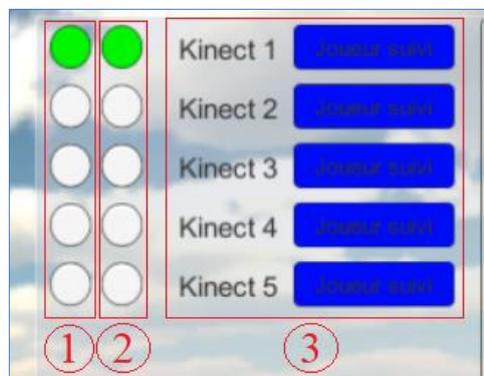


Figure 4.33 : Liste des états de suivis des capteurs

3.2.4 Le panneau « Consulter Examen »

Le panneau « Consulter examen » du menu principal, illustré dans la figure 4.34, permet d’afficher une interface contenant la liste de tous les patients déjà sauvegardés (voir figure 4.28).

De cette interface (figure 4.34) le médecin/utilisateur sélectionne le patient souhaitant consulter ses examens.



Figure 4.34 : Panneau Consulter Examen

La liste des patients

ID	Nom	Prenom	Age	Sexe	Situation	Tel
13	Djekoune	Walid	39	Homme	Marié(e)	0777586932
14	Ramdane	Souhaib	23	Homme	Célibataire	0555695847
16	Zola	Lella	43	Femme	Marié(e)	0791253614
17	Hadjersi	Mohammed	24	Homme	Célibataire	0668241532
18	Gizelaine	Sara	28	Femme	Célibataire	0569878421
19	Kallia	Anis	14	Homme	Célibataire	0663214596
20	Ousalah	Ihcen	25	Femme	Célibataire	0669632001
21	Osmane	Malika	29	Femme	Marié(e)	0687695874

* Pour consulter les examens d'un patient, sélectionnez la ligne dans le tableau Patient

Figure 4.35 : Interface de sélection du patient pour la consultation d'examen

En cliquant sur le bouton « Continuer », une interface contenant les informations du patient sélectionné ainsi que tous ses examens effectués sont affichés (voir figure 4.36). Ceci permet au médecin/utilisateur de sélectionner un examen souhaitant le consulter ou le supprimer.

Information patient

Nom : Ramdane
 Prénom : Souhaib
 Date de naissance : 9/13/1996
 Sexe : Homme
 Situation familiale : Célibataire
 Adresse : Cité Iskander Bachir N1 Blida
 Tel : 0555695847

ID	Date	Heure	Superviseur
7	8/22/2020	16:25:11	Dr. Anane Imad
8	8/22/2020	16:28:9	Dr. Anane Imad
9	8/22/2020	16:30:2	Dr. Zitouni Malika

* Pour consulter ou supprimer un examen, sélectionnez la ligne dans le tableau Examen

Figure 4.36 : Interface des informations du patient et ses examens

Dans le cas d'une consultation d'examen, l'interface illustrée dans la figure 4.37 est affichée. Elle contient toutes les informations concernant un examen à savoir :

- Informations sur le patient : nom, prénom, sexe, date de naissance, maladie, téléphone.
- Informations sur l'examen : la date, l'heure, le médecin superviseur, la durée de l'animation ainsi que le chemin du fichier CSV contenant toutes les données squelettiques de l'animation lors de la séance de l'examen.
- Les notes enregistrées lors de la séance d'examen.

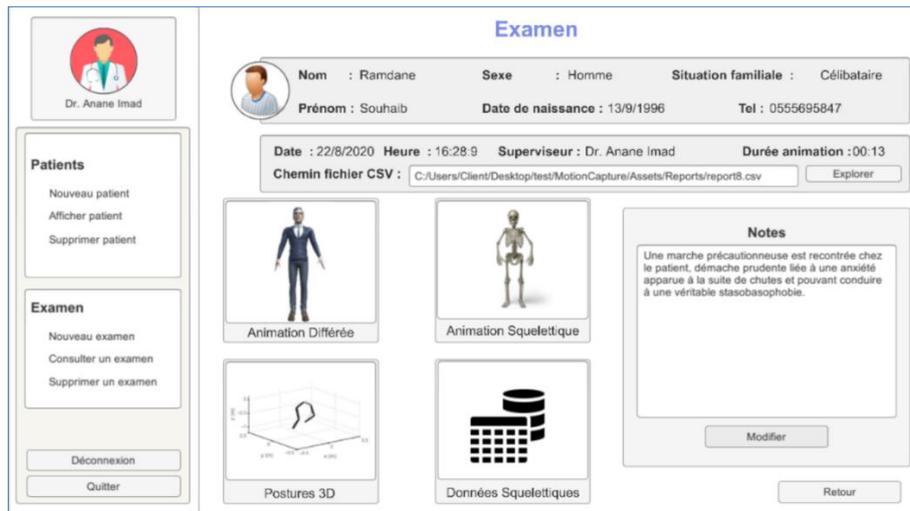


Figure 4.37 : Interface consultation d'un examen

L'interface contient aussi un ensemble de boutons fournissant des fonctionnalités importantes, à savoir :

- Le bouton « Explorer » : permet d'ouvrir directement le fichier CSV via l'application Excel de Microsoft contenant toutes les données squelettiques 3D des articulations de chaque frame des mouvements effectués par le patient comme le montre la figure 3.38.

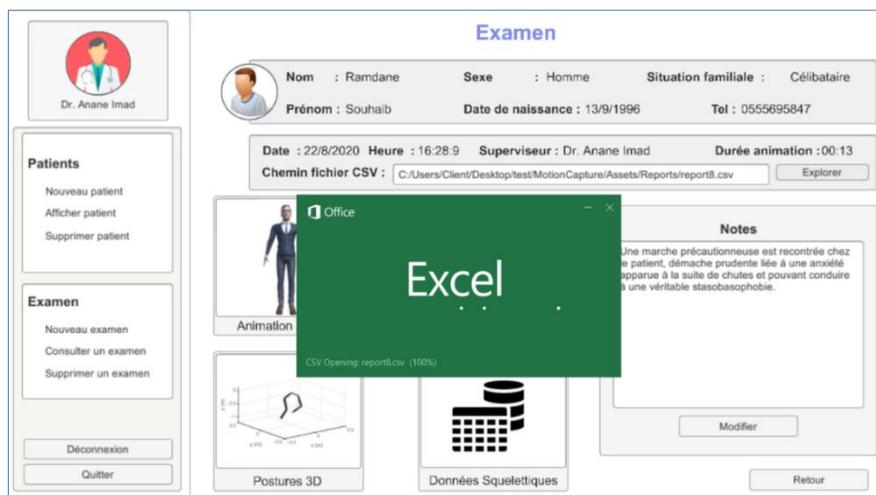


Figure 4.38 : Interface de sélection du patient pour la consultation d'examen

- Le bouton « Modifier » permet de modifier les notes de l'examen courant.
- Le bouton « Données Squelettiques » : permet d'afficher les données squelettiques 3D des articulations des mouvements effectués par un patient dans un tableau bien structuré (voir la figure 4.39).

Examen

Chemin fichier CSV : C:/Users/Client/Desktop/test/MotionCapture/Assets/Reports/report8 Ouvrir

Frame	HipCenter			Spine			ShoulderCenter			Head		
	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
Frame 1	-0.3573323	0.0252485	2.537689	-0.3793592	0.0884027	2.595025	-0.4202023	0.4645277	2.596138	-0.4059575	0.6524014	2.5622
Frame 2	-0.3605948	0.0270346	2.53589	-0.3812486	0.0895571	2.593331	-0.4211333	0.4640776	2.594508	-0.4054594	0.6509838	2.5595
Frame 3	-0.3660342	0.0262054	2.531671	-0.3849798	0.0887587	2.589285	-0.4255411	0.4614874	2.590323	-0.4169734	0.6452253	2.5460
Frame 4	-0.3714247	0.0232290	2.526021	-0.3910293	0.0858688	2.583007	-0.4334795	0.4590746	2.584429	-0.4270453	0.6428862	2.5370
Frame 5	-0.3771617	0.0235447	2.519037	-0.398465	0.0862264	2.575032	-0.4413381	0.4591456	2.575805	-0.4297606	0.6429669	2.5340
Frame 6	-0.3816601	0.0233458	2.514172	-0.4067962	0.0858303	2.568849	-0.4492695	0.4610064	2.572366	-0.4322802	0.6434594	2.5311
Frame 7	-0.3864506	0.0212508	2.508676	-0.414574	0.0839162	2.562812	-0.4561506	0.4605147	2.565265	-0.4350345	0.6440724	2.5282
Frame 8	-0.3910407	0.0176125	2.502764	-0.420817	0.0809859	2.555981	-0.4593175	0.4563627	2.550277	-0.4478755	0.6450033	2.5152
Frame 9	-0.3926944	0.0159738	2.500588	-0.4232419	0.0797374	2.553101	-0.4606549	0.4558387	2.544805	-0.44919	0.6456508	2.5129
Frame 10	-0.3931117	0.0154846	2.500008	-0.4241588	0.0793187	2.552099	-0.4609499	0.45587	2.543765	-0.4488553	0.6459544	2.5128
Frame 11	-0.3933102	0.0152302	2.499716	-0.4246795	0.0791111	2.551547	-0.4610943	0.4560389	2.543423	-0.448424	0.6463929	2.5136
Frame 12	-0.3934119	0.0151327	2.499688	-0.424952	0.0790582	2.551404	-0.4612772	0.4563132	2.543436	-0.4477701	0.6475168	2.5155
Frame 13	-0.3936115	0.0151466	2.50014	-0.424793	0.0792303	2.551987	-0.4613233	0.4566754	2.544194	-0.4481109	0.6479582	2.5169
Frame 14	-0.3942967	0.0161420	2.501247	-0.4238811	0.0805886	2.553947	-0.4611524	0.4591005	2.548616	-0.4487917	0.6480153	2.5174
Frame 15	-0.3953121	0.0172048	2.502022	-0.4232988	0.0817279	2.555274	-0.4611511	0.460324	2.550747	-0.4500048	0.6480705	2.5167

Retour

Figure 4.39 : Interface Tableau des données squelettiques 3D d'une animation

- Le bouton « Animation Différée » permet de revoir des examens antérieurs de patients, en récupérant les données squelettiques 3D du fichier JSON de l'animation live comme le montre la figure 4.40. Le médecin peut basculer dans la scène virtuelle à travers les différentes touches du clavier spécifiées en bas de l'écran.

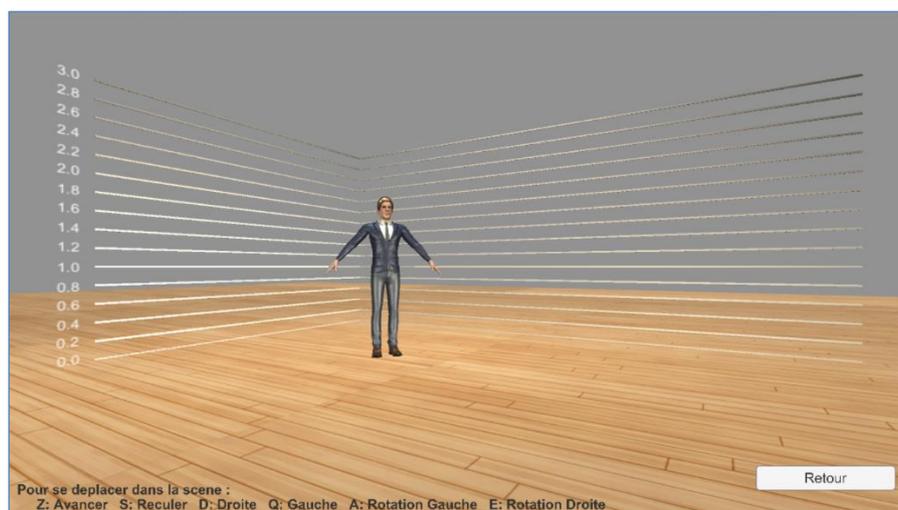


Figure 4.40 : Scène de l'animation différée

- Le bouton « Postures 3D » permet au médecin/utilisateur de visualiser les différentes postures 3D du patient de chaque frame de l’animation virtuelle différée. Il lui permet en premier lieu de sélectionner les parties du corps que le médecin/utilisateur souhaite visualiser en cliquant sur les boutons du panneau gauche de l’interface ou en sélectionnant directement les articulations du modèle squelettique via l’interface illustrée dans la figure 4.41.

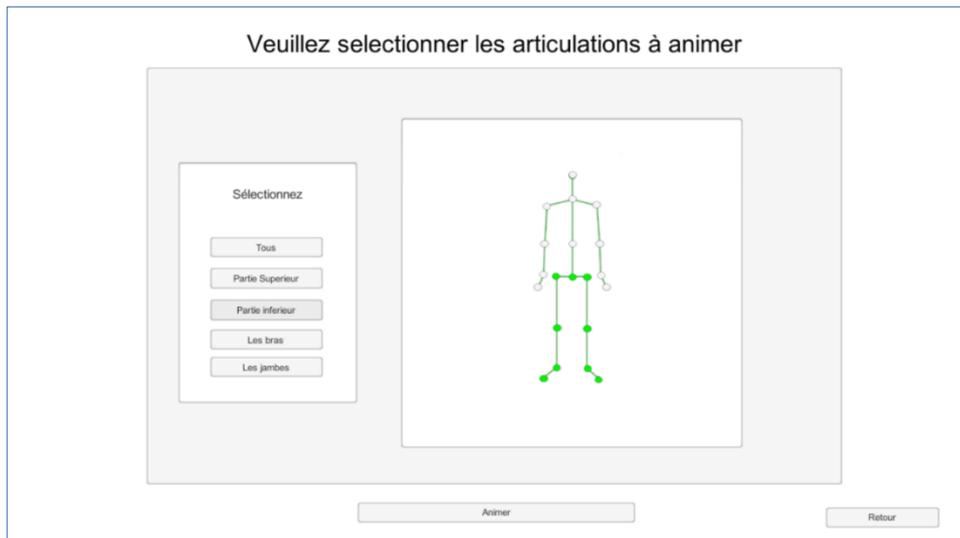


Figure 4.41 : Interface de sélection des articulations pour la représentation 3D des postures

Après la sélection des articulations, la scène des postures de chaque frame est lancée en représentation 3D où le médecin/utilisateur peut aussi basculer en cliquant sur les touches du clavier spécifiées en bas de l’écran comme le montre la figure 4.42.

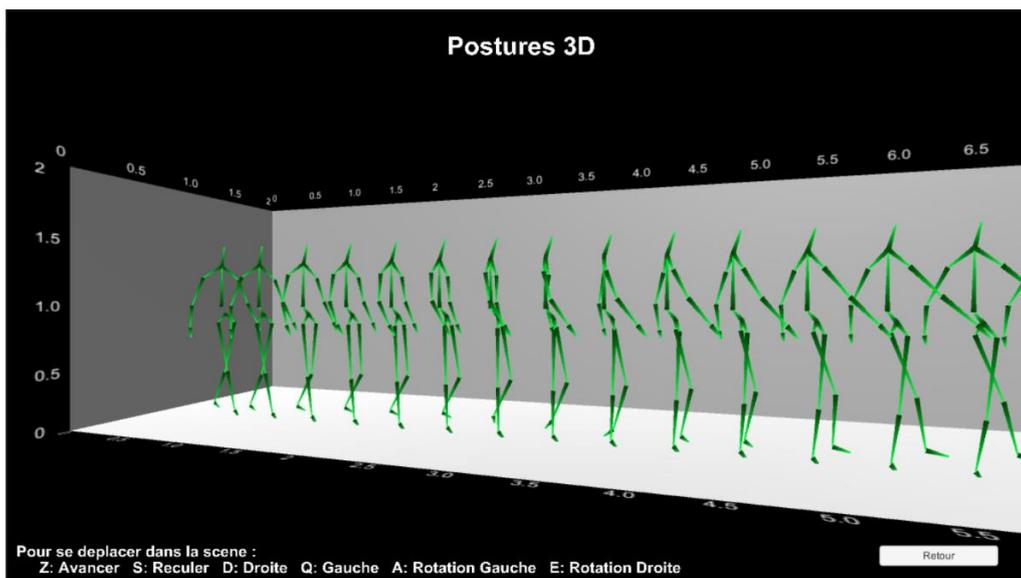


Figure 4.42 : Scène de représentation 3D des postures

- Le bouton « Animation Squelettique » permet d’animer un squelette avec les données 3D enregistrées de l’animation live dans une scène virtuelle afin d’offrir au médecin/utilisateur un moyen de visualiser les différents translations et rotations de chaque articulation (voir figure 4.43).

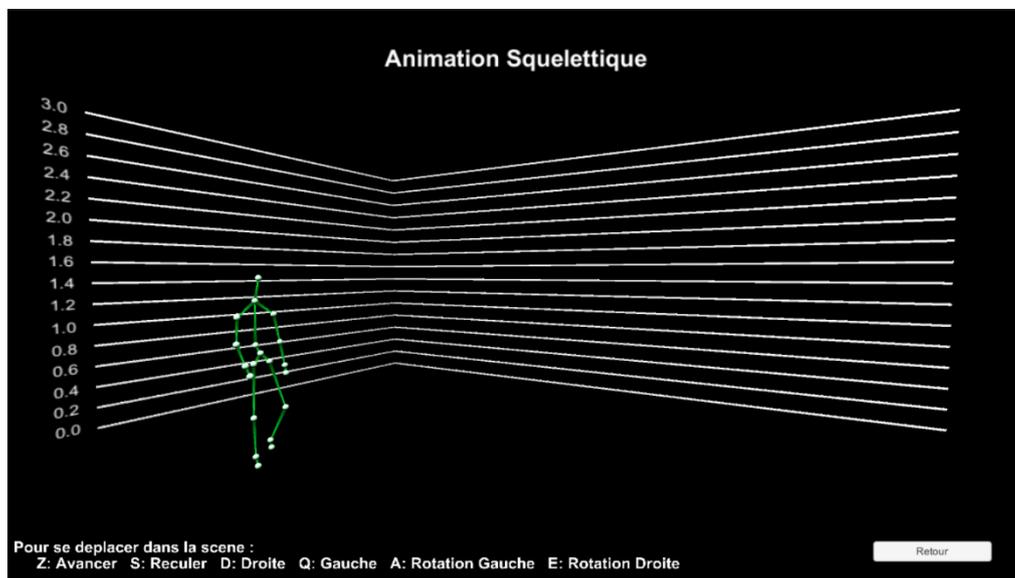


Figure 4.43 : Scène d’animation squelettique

4 Tests et évaluations

4.1 Configuration du système développé

Afin de déterminer l’importance/influence du nombre des capteurs utilisés dans notre système sur le résultat de la capture de mouvements réels et l’animation virtuelle de l’avatar, nous avons élaboré une série de tests avec des configurations différentes en utilisant :

4.1.1 Configuration avec un seul capteur

Les limites dans cette configuration avec un seul capteur Kinect, sont, en premier lieu, dues au fait que le champ de capture de mouvements est sur un angle de « 180° » en premier lieu. Ce qui limite la liberté de déplacement du sujet humain dans la scène. En second lieu, l’auto-occultation des membres du corps est fréquemment produite lorsque le sujet est de profile par rapport au capteur Kinect ou croise ses membres. Ce qui entraîne l’absence ou la non précision des données concernant les articulations occultées. La figure 4.44 montre de l’Avatar dans le monde virtuel ayant la même posture du sujet humain face à l’unique capteur Kinect utilisé.

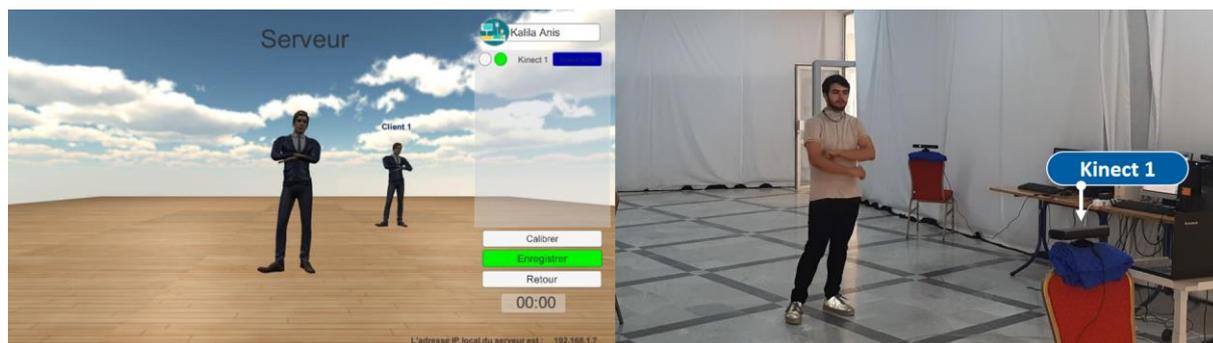


Figure 4.44 : Configuration avec un seul capteur

4.1.2 Configuration avec deux capteurs

Dans cette configuration, nous plaçons les capteurs à une distance d'environ de trois mètres (portée max du capteur) de telle sorte à former un angle entre 60° et 90° . Cette disposition nous permet d'étendre le champ horizontal de la capture de mouvements en fournissant au sujet humain une distance de deux mètres de déplacement avec une rotation de 180° faisant face à un capteur Kinect ou à deux en même temps. Notre approche de fusion des données squelettiques reconstruit la posture humaine en corrigeant à chaque frame les articulations non traquées ou inférées par les capteurs. Néanmoins, cette configuration limite la rotation du sujet humain à 90° degré à droite et à gauche. La figure 4.45 montre bien les Avatars des deux machines clients ainsi que l'Avatar reconstruit ayant la même posture du sujet humain entre deux capteurs Kinect utilisés et dont le visage n'est face à aucun d'eux.

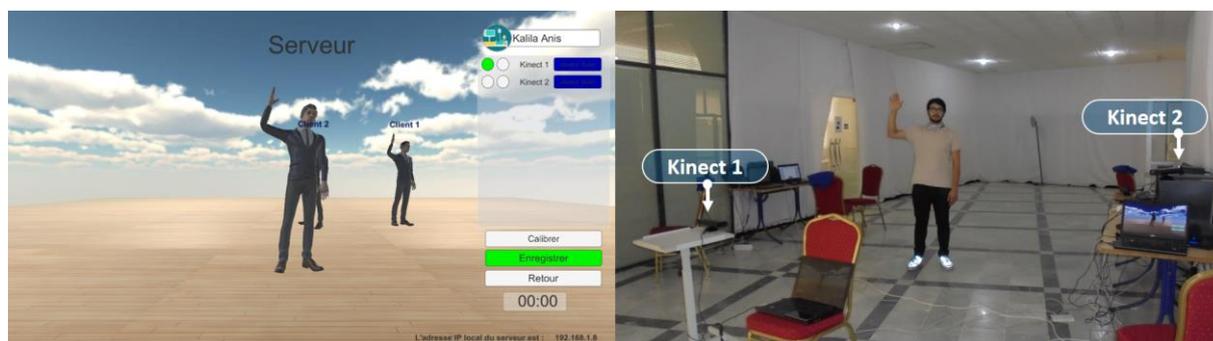


Figure 4.45 : Configuration avec deux capteurs

4.1.3 Configuration avec trois capteurs

Dans le cas de la configuration avec trois capteurs, nous les plaçons sous forme d'un triangle avec un angle d'environ de 120° entre deux axes de vue. Cette configuration nous offre un champ de capture plus vaste que les configurations précédentes en élargissant la liberté de déplacement et de rotation du sujet humain dans la scène. Une amélioration remarquable sur le réalisme de l'animation résultant de la fusion des données squelettiques provenant de plusieurs

sources avec des angles de vue différents. Cependant, le champ de liberté dans la scène reste restreint dans cette configuration. La figure 4.46 montre les trois Avatars reconstruits par les trois machines clients ainsi que l'Avatar reconstruit ayant la même posture du sujet humain entre trois capteurs Kinect utilisés et dont le visage n'est face à aucun d'eux.



Figure 4.46 : Configuration avec trois capteurs

4.1.4 Configuration avec quatre capteurs

Dans le cas de la configuration avec quatre capteurs, nous les plaçons sur les sommets d'un carré/rectangle où chaque paire de capteurs Kinect se place face. L'avantage principal de cette configuration est la couverture complète de la zone de la capture de mouvements en offrant au sujet humain une liberté totale de rotation de 360° dans la scène. Cette configuration donne des meilleurs résultats de suivi des séquences des mouvements du sujet humain. La figure 4.47 montre les quatre Avatars reconstruits par les quatre machines clients ainsi que l'Avatar reconstruit ayant la même posture du sujet humain entre quatre capteurs Kinect utilisés et dont le visage est face au capteur 2.



Figure 4.47 : Configuration avec un quatre capteurs

4.1.5 Configuration avec cinq capteurs

La configuration avec cinq capteurs est représentée sur la figure 4.48 où les capteurs ont été placés aux sommets d'un pentagone avec des côtés d'environ de 3 mètres de longueur.

Idéalement, cette configuration nous donne les meilleurs résultats de capture et de suivi des mouvements exercés par le sujet humain avec une absence totale des problèmes liés aux occultations et aux limitations du degré de rotation dans la scène réelle, en plus d'une augmentation considérable dans la précision et le réalisme de l'animation virtuelle issus de la fiabilité des données squelettiques résultantes de fusion des données provenant des cinq capteurs. La figure 4.48 montre les cinq Avatars reconstruits par les cinq machines clients ainsi que l'Avatar reconstruit ayant la même posture du sujet humain entre cinq capteurs Kinect utilisés et dont le visage est face au capteur 1. Cette figure montre aussi les données erronées envoyées par le capteur 3 vers la machine serveur, l'Avatar est de face tandis que le sujet humain est de dos par rapport à ce capteur. La machine serveur a corrigé ces erreurs en se basant sur les données en provenance des autres capteurs.

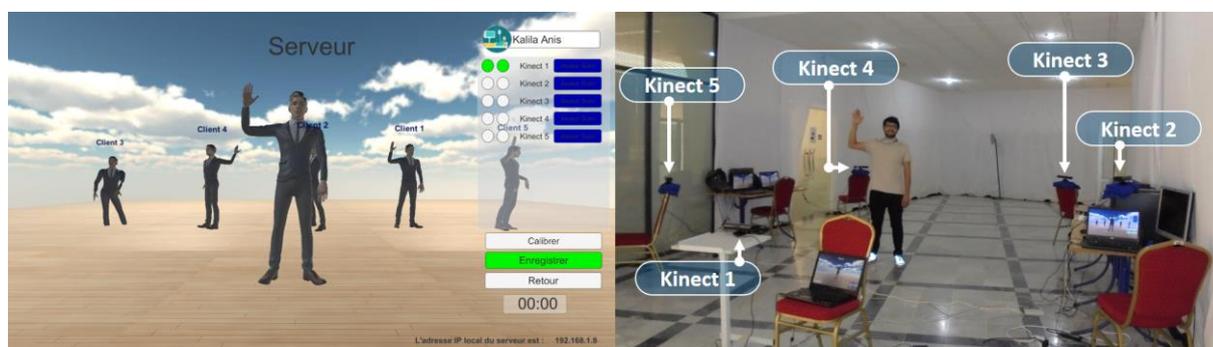


Figure 4.48 : Configuration avec cinq capteurs

4.2 Etude expérimentale

L'objectif fondamental de notre système est de reproduire le plus fidèlement possible les mouvements exercés par le sujet humain dans un environnement virtuel par un Avatar en temps réel et d'une manière naturelle tout en garantissant une animation réaliste et précise.

L'utilisation des capteurs Kinect souffrent fréquemment des problèmes d'occultations et surtout lorsqu'ils sont utilisés seuls. Comme nous pouvons le voir dans le cas « A » et « B » de la figure 4.49, le sujet humain est face au capteur 4 où nous remarquons une déformation presque totale des Avatars affichés par les machines clients 3 et 5. Cela revient principalement aux occultations de la majorité des articulations du côté gauche pour le capteur 3 et le côté droit pour le capteur 5 du sujet humain.

Le même problème soulevé dans la figure 4.48 est présent dans cette même figure. Il s'agit du problème où le dos du sujet humain est face au capteur, cas des capteurs 1 et 2. Nous remarquons bien que les Avatars des machines clients 1 et 2 sont montrés de face (voir la zone « C » de la même figure). Le capteur possède l'incapacité de distinguer entre une capture de

face et de dos du sujet humain. Contrairement, une stabilité et un réalisme totale est remarqué sur la posture de l'avatar fusionné en comparant avec celle du sujet humain.



Figure 4.49 : Expérience de configuration avec cinq capteurs

Selon l'expérience réalisée en utilisant la configuration, de cinq capteurs, illustrée dans la figure 4.49, nous pouvons distinguer les faits suivants :

- En premier lieu, une élimination totale du problème d'occultation dans l'avatar fusionné résultant de l'approche de fusion des données squelettiques provenant des différents capteurs. Ce qui élimine l'incertitude des données inférées ou non traquées et qui augmente leur fiabilité.
- En second lieu, une couverture de 360° du sujet humain, ce qui lui confère une liberté de rotation et de déplacement dans toutes les directions sans inférence sur l'animation dans le monde virtuel.
- En troisièmes lieu, notre approche de détection du visage élimine radicalement le problème de l'incapacité du capteur à distinguer entre la capture de face et de dos, ce qui assure l'exactitude du choix du squelette de base à chaque frame.

- En dernier lieu, une animation réaliste, naturelle et fluide qui reflète exactement les mouvements réels exercés par le sujet humain tout en respectant l'aspect « temps réel » de l'animation dans le monde virtuel.

La figure 4.50 représente un exemple d'une capture d'un mouvement humain, son animation en temps réel et son animation en mode différée (Avatar/squelette).

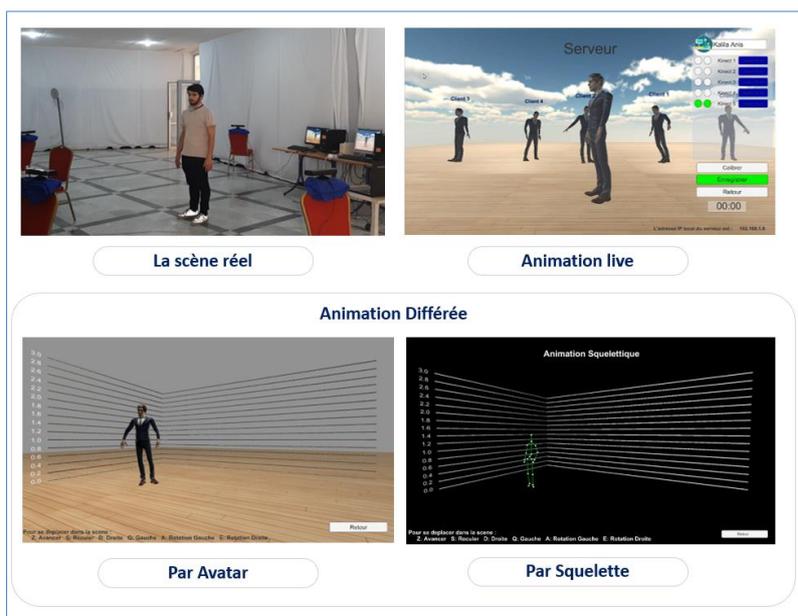


Figure 4.50 : Exemple d'animation d'un mouvement

4.2.1 Sauvegarde des données squelettiques

Le médecin/utilisateur peut enregistrer l'animation de l'examen effectué par le patient en cliquant sur le bouton « Enregistrer » de l'interface d'animation virtuelle. Le produit final de ce processus est un fichier « JSON » contenant toutes les données squelettiques fusionnées de tous les frames de la séance d'examen. Ces données sont exploitées lors de l'animation différée de ce même examen. Un exemple de ce fichier est illustré dans la figure 4.51.

```

1  {"AnimationData": {
2    "articuPosition":
3    [
4      {"x":0.0000266947315940633,"y":0.9997748136520386,"z":0.004547574557363987},
5      {"x":0.0,"y":0.09694427251815796,"z":-0.011097324000765248},
6      {"x":0.0,"y":0.1463659586734793,"z":-2.3259993203597508e-03},
7      {"x":0.0,"y":0.06944289003504944,"z":0.019824808463454248},
8      {"x":0.06681303679943085,"y":0.12887641787528993,"z":0.0015732940519228578},
9      {"x":-4.044393292490511e-11,"y":0.1401614397764206,"z":4.1285630469900527e-10},
10     {"x":-2.2562728074770889e-10,"y":0.2733026146888733,"z":-5.332679783864869e-11},
11     {"x":0.546457374605574e-11,"y":0.20888808369636537,"z":3.664894221433457e-10},
12     {"x":0.035568881775009137,"y":0.11070999013015562,"z":0.001536398078314961},
13     {"x":0.06681303679943085,"y":0.1289733052253732,"z":0.0024197502061724664},
14     {"x":9.148960755656077e-11,"y":0.1401614397764206,"z":4.067513270644696e-10},
15     {"x":-3.0047878074599053e-10,"y":0.2731824517250061,"z":-3.198641629342802e-10},
16     {"x":3.802657908146827e-11,"y":0.20889824628929957,"z":5.525663260286251e-11},
17     {"x":0.03650906364046922,"y":0.11210109293460896,"z":0.0000199461728170514},
18     {"x":0.009726321905851364,"y":0.05393180623650551,"z":0.004454033440292183},
19     {"x":-1.0183850429612563e-9,"y":0.4105736017227173,"z":-7.25783183552764e-11},
20     {"x":-5.666295765927032e-9,"y":0.4203698933124542,"z":-5.308817274674027e-10},
21     {"x":-3.4968272718692e-10,"y":0.1868141293526959,"z":-7.826384873510506e-9},
22     {"x":0.0874321905851364,"y":0.05393180623650551,"z":0.004454033440292183},
23     {"x":2.507781760030525e-10,"y":0.41054293513280037,"z":3.5567865650624869e-10},
24     {"x":2.2851821823710595e-10,"y":0.41574708437919619,"z":1.189995797142249e-10},
25     {"x":0.243082882941621e-10,"y":0.1821046769618988,"z":5.342026265796615e-10}
26   ],
27   "articuRotation":
28   [
29     {"x":0.005791701842099428,"y":0.05309250205755234,"z":0.0016968180425465108,"w":0.9985713958740234},
30     {"x":0.006129629164934158,"y":0.04558559879660607,"z":0.02832607820630075,"w":0.996675509376526},
31     {"x":0.21606913208961488,"y":0.021784890443096625,"z":0.1313645541667993,"w":0.9672554135322571},
32     {"x":0.0032937671057879926,"y":0.000611809955441869,"z":0.01221483212351799,"w":0.9999198913574219},
33     {"x":0.5514443518731262,"y":0.4392883506011983,"z":0.5642642974939516,"w":0.4296028338119505},
34     {"x":0.35485145449638369,"y":0.12734517455101014,"z":0.13476136326789857,"w":0.9163531064987183},
35     {"x":0.4791932702064514,"y":0.12879283726215363,"z":0.6017656922340393,"w":0.6258308294435242},
36     {"x":0.03227823431921008,"y":0.03167455270886421,"z":0.08133576065301895,"w":0.9956602454185486},
37     {"x":0.0008439457742497325,"y":0.005929636303335428,"z":0.049415787518024448,"w":0.9988062780113322},
38     {"x":0.547592328915496,"y":0.442311793916110537,"z":0.5874098845478565,"w":0.42727678795890245},
39     {"x":0.338474464635110476,"y":0.03623668682935927,"z":0.09720484167337418,"w":0.9171749945450261},
40     {"x":0.014231132000281048,"y":0.5242899159303894,"z":0.33300480246543887,"w":0.7935959792137146},
41     {"x":0.045349203050136569,"y":0.09006983041763306,"z":0.0036276322789490225,"w":0.9948988158493042},
42     {"x":0.00162041524899081026,"y":0.009048616513609887,"z":0.0456253029460707,"w":0.9989163279533386},
43     {"x":0.024013616144657136,"y":0.008557276800274849,"z":0.9569538450241089,"w":0.07370974868535996},
44   ]
45 }

```

Figure 4.51 : Exemple d'un fichier « JSON » des données d'une animation

Un autre fichier « CSV » est produit, contenant les positions 3D de toutes les articulations dans chaque frame de l'animation afin d'être visualisé et exploité par d'autres logiciels de représentation graphique (voir figure 4.52).

```

1  Frames;HipCenter X;HipCenter Y;HipCenter Z;Spine X;Spine Y;Spine Z;ShoulderCenter X;ShoulderCenter Y;ShoulderCenter Z;Head X;Head
2  Frame 1;-0.5041769;0.1055589;2.427834;-0.5123619;0.1931372;2.429296;-0.5382544;0.6297888;2.444333;-0.4675629;0.7984871;2.46856;-0
3  Frame 2;-0.4693139;0.1007647;2.423392;-0.4776781;0.1883335;2.426394;-0.507281;0.6264311;2.449554;-0.4314984;0.7951416;2.474153;-0
4  Frame 3;-0.364779;0.1273565;2.436241;-0.3737496;0.2105508;2.439572;-0.4012264;0.6326718;2.464456;-0.3322238;0.7993063;2.483321;-0
5  Frame 4;-0.3363471;0.1376214;2.438632;-0.3469063;0.2192782;2.442025;-0.3841017;0.6341074;2.466607;-0.3118896;0.8016781;2.485744;-0
6  Frame 5;-0.3240447;0.1392878;2.440329;-0.3332502;0.2206711;2.443652;-0.3619872;0.6323376;2.466125;-0.290429;0.801021;2.485991;-0
7  Frame 6;-0.2843637;0.1777513;2.454453;-0.2980833;0.2473948;2.453802;-0.3422273;0.6304471;2.467453;-0.2677226;0.8004598;2.48843;-0
8  Frame 7;-0.2843637;0.1777513;2.454453;-0.2980833;0.2473948;2.453802;-0.3422273;0.6304471;2.467453;-0.2677226;0.8004598;2.48843;-0
9  Frame 8;-0.2704659;0.1933335;2.457077;-0.2805299;0.2642246;2.457093;-0.3131007;0.6309974;2.469985;-0.2391149;0.8026914;2.485665;-0
10 Frame 9;-0.2704659;0.1933335;2.457077;-0.2805299;0.2642246;2.457093;-0.3131007;0.6309974;2.469985;-0.2391149;0.8026914;2.485665;-0
11 Frame 10;-0.2670777;0.1940063;2.455796;-0.2766177;0.2653121;2.456401;-0.3084644;0.6310599;2.470681;-0.2321714;0.8036086;2.482773;-0
12 Frame 11;-0.2611064;0.1909567;2.452067;-0.2705548;0.2636316;2.454038;-0.2987658;0.6309072;2.471092;-0.2270058;0.8041594;2.48021;-0
13 Frame 12;-0.2532101;0.1893143;2.44783;-0.2623598;0.2626369;2.450847;-0.2911438;0.6308095;2.471088;-0.2168712;0.8035245;2.475684;-0
14 Frame 13;-0.2441579;0.1872758;2.443922;-0.2526105;0.260891;2.44759;-0.2811277;0.629773;2.470596;-0.2074016;0.8024468;2.472186;-0
15 Frame 14;-0.2326712;0.1892591;2.44052;-0.2414304;0.2620303;2.444464;-0.2711737;0.6284477;2.469351;-0.196334;0.7994745;2.467318;-0
16 Frame 15;-0.2239869;0.1899182;2.437574;-0.2326535;0.2621487;2.441939;-0.2625711;0.6265193;2.467847;-0.178928;0.7931076;2.460339;-0
17 Frame 16;-0.2239869;0.1899182;2.437574;-0.2326535;0.2621487;2.441939;-0.2625711;0.6265193;2.467847;-0.178928;0.7931076;2.460339;-0
18 Frame 17;-0.2160847;0.1910189;2.434584;-0.2244673;0.2626042;2.43921;-0.2518728;0.6231995;2.466549;-0.169918;0.7901429;2.456296;-0
19 Frame 18;-0.1959018;0.1876254;2.420543;-0.2052636;0.2587461;2.42765;-0.2240604;0.6151032;2.453955;-0.1524294;0.7858902;2.446939;-0
20 Frame 19;-0.1849972;0.1837766;2.411685;-0.196948;0.2551669;2.4219;-0.211105;0.6117671;2.446609;-0.1402852;0.784005;2.43987;-0.318
21 Frame 20;-0.1849972;0.1837766;2.411685;-0.196948;0.2551669;2.4219;-0.211105;0.6117671;2.446609;-0.1402852;0.784005;2.43987;-0.318
22 Frame 21;-0.163938;0.1789463;2.403206;-0.1827517;0.2506433;2.416904;-0.201811;0.6100839;2.440958;-0.1288395;0.782199;2.432164;-0

```

Figure 4.52 : Exemple d'un fichier « CSV » des données d'une animation

4.3 Evaluation quantitative du système développé

Pour une évaluation quantitative de notre système de capture de mouvement et d'animation virtuelle, nous effectuons dans les différentes configurations citées précédemment une méthode permettant d'enregistrer pour 2000 frames :

- Le temps d'exécution de la fusion et d'animation des données provenant des différents capteurs.
- Round-Time-Trip (RTT), qui fait référence aux durées du trajet aller-retour dans le réseau lors de l'envoi des données squelettiques d'un client au serveur.

D'après le graphe illustré dans la figure 4.53, nous constatons que les temps d'exécution moyens par frame de l'algorithme de fusion et de l'animation de l'avatar que nous avons développé évoluent à chaque fois que nous augmentons le nombre de capteurs Kinect dans notre système de capture des mouvements. D'où une « Relation proportionnelle » entre le nombre de capteurs utilisés et le temps d'exécution de l'algorithme de fusion et d'animation. En revanche, l'aspect temps réel de l'animation virtuelle reste toujours respecté. De ce fait, le meilleur compromis que nous avons trouvé entre le temps d'exécution qui nous assure l'aspect temps réel et la configuration de Kinect qui nous fournit une capture de mouvements performante et précise est celle où nous utilisons cinq capteurs Kinect.

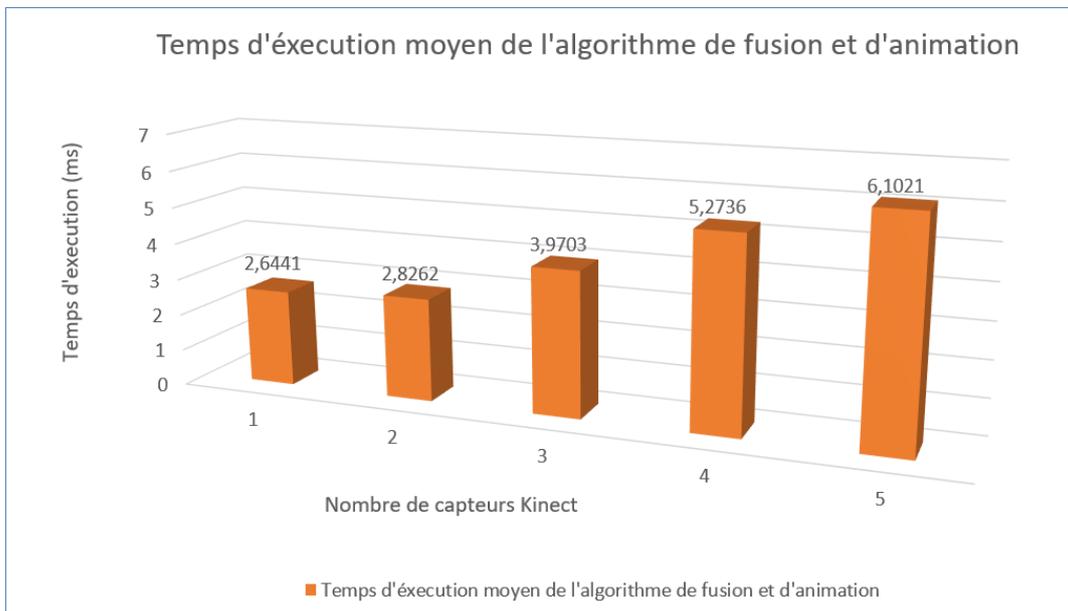


Figure 4.53 : Temps d'exécution de l'algorithme de fusion développé et animation en fonction du nombre des capteurs utilisés

De même, nous remarquons que le RTT du réseau qui concerne le temps de trajet que prennent les données squelettiques pour être transmises du client au serveur augmente légèrement à chaque ajout d'un capteur Kinect ou d'un client au réseau comme montre la figure 4.54.

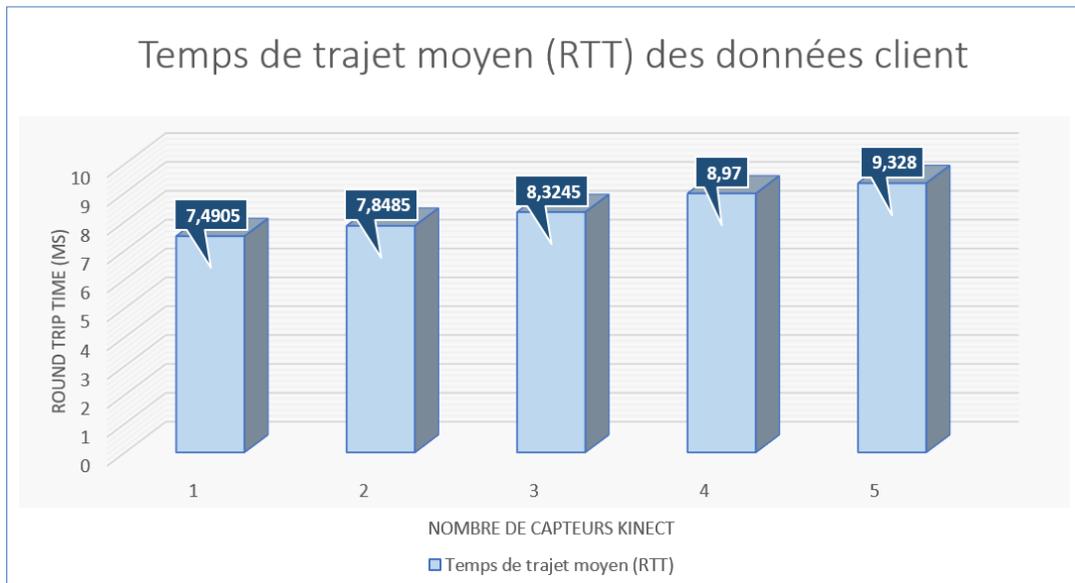


Figure 4.54 : Temps de trajet moyen (RTT) des données client en fonction du nombre des capteurs utilisés

5 Conclusion

Dans le présent chapitre, nous avons tout d'abord présenté les différentes technologies et les outils que nous avons utilisés pour l'implémentation de notre système de capture de mouvements. Nous avons par la suite présenté les principales fonctionnalités de notre application qui gèrent les données issues de notre système ainsi que ses différentes interfaces et scènes virtuelles. Puis, nous avons détaillé les différentes configurations utilisées en fonction du nombre de capteurs utilisés pour le test et l'évaluation de notre approche. Une étude expérimentale et une évaluation quantitative ont été présentées en dernier lieu afin de dégager ses points forts et ses limites. Les tests effectués confirment que le système que nous avons développé offre des résultats très satisfaisants, offrant d'une part une capture des mouvements précise et performante et d'autre part une animation d'avatar naturelle, réaliste et fluide avec un rendu visuel en temps réel. De plus, nous pouvons conclure que notre approche de fusion de données squelettiques multi capteurs apporte une nette amélioration en temps et en réalisme de l'animation lorsque nous utilisons un nombre important de capteurs Kinect. Quant au coût de calcul ou de la latence du système, elle reste fort acceptable et négligeable pour une application de réalité virtuelle destinée à la rééducation fonctionnelle de la marche. Ceci nous amène au terme de ce chapitre, et à la conclusion de ce travail

Conclusion Générale

Le présent travail est une contribution multidisciplinaire qui touche plusieurs domaines de la recherche scientifique, à savoir : la réalité virtuelle, la vision artificielle, l'IHM (interaction Homme/Machine), la recherche clinique et la réadaptation fonctionnelle. Il consiste en premier lieu à la mise en œuvre d'une plateforme multi capteurs Kinect fournissant une capture totale du mouvement du corps d'un sujet humain avec une couverture angulaire de 360°. Une architecture Client/Serveur a été proposée à cet effet pour collecter les données squelettiques des machines clients et les transmettre en temps réel au serveur pour la reconstruction finale du modèle du squelette du sujet humain dans un monde virtuel. Il comporte en deuxième lieu une approche mathématique pour le calibrage permettant de trouver les correspondances et les relations fiables et précises entre les différents repères des capteurs utilisés afin de fournir une fusion homogène des données squelettiques. L'approche de la fusion des données a permis de reconstruire un squelette robuste du sujet humain contenant que des articulations ayant des informations exactes et précises. De plus, ce travail assure une animation naturelle, réaliste et précise des Avatars dans un environnement virtuel interactif qui reflète le plus fidèlement la réalité. Bien que notre travail soit particulièrement destiné, dans un premier temps, au domaine de la médecine à savoir la rééducation fonctionnelle pour des patients ayant des difficultés à marcher, il est aussi extensible à d'autres domaines et applications qui font interagir des Avatars du monde virtuel par des humains du monde réel.

Les objectifs atteints avec succès au cours de ce travail sont :

- La conception et la réalisation d'un système d'aide au diagnostic et au suivi des patients atteints de pathologies dans un cadre de rééducation et de réadaptation locomotrice et neurologique.
- L'utilisation d'un matériel grand public, ce qui confère à notre système la particularité d'être bon marché.
- L'adoption de l'approche client/serveur dans la conception de notre système qui a permis de bien distribuer les tâches entre les machines et d'atteindre le temps réel dans les traitements et la visualisation des examens.
- Le choix de l'utilisation de cinq capteurs Kinect représente le meilleur compromis entre la précision, le réalisme de l'animation et le traitement en temps réel.

- L'élimination du problème d'occultation grâce à notre approche de fusion des données squelettiques qui corrige les données manquantes provenant d'un seul capteur Kinect du réseau.
- La couverture de 360° du corps du sujet humain qui élimine l'obligation d'être toujours face au capteur tout au long de la capture de mouvements et donne une liberté totale de déplacement et de rotation dans la scène.
- La résolution du problème lié à la non distinction entre la vue de face et celle de dos du sujet humain grâce à notre algorithme de fusion de données
- Un moyen de suivi et de gestion des sujets humains avec la possibilité de visualiser en mode différé les mouvements enregistrés, des examens antérieurs.
- Une présentation graphique 3D des données squelettiques de chaque articulation ainsi que les postures 3D à chaque frame, qui permet au médecin traitant la visualisation des anomalies des postures des patients lors des examens.
- La génération des fichiers JSON et CSV contenant les données squelettiques 3D fusionnées. Ceux-ci peuvent faire l'objet des données d'entrées pour le développement d'outils d'aide au diagnostic automatique.

Perspectives

Afin d'améliorer au mieux notre système, nous avons dégagé certaines perspectives futures à considérer, à savoir :

- L'utilisation de la deuxième version des capteurs Kinect : Kinect Xbox One V2 au lieu de ceux que nous avons utilisés. Ils offrent une stabilité remarquable du suivi du squelette du sujet humain avec une meilleure précision des données squelettiques renvoyées (25 articulations au lieu de 20). Ces capteurs possèdent aussi un champ de vision horizontalement et verticalement plus large ce qui permet d'étendre la zone de la capture de mouvements réels.
- Les données squelettiques résultantes de notre système de fusion et d'animation virtuelle peuvent être analysées par l'intégration d'un sous-système d'aide au diagnostic automatique qui permet d'identifier avec précision les anomalies et les problèmes neurologiques dans les mouvements en se basant sur les techniques d'apprentissage profond.

Bibliographie

- [1] CAPPOZZO, Aurelio, DELLA CROCE, Ugo, LEARDINI, Alberto, et al. "Human movement analysis using stereophotogrammetry: Part 1: theoretical background". *Gait & posture*, 2005, vol. 21, no 2, p. 186-196.
- [2] LI, Shun, CUI, Liqing, ZHU, Changye, et al. "Emotion recognition using Kinect motion capture data of human gait"s. *PeerJ*, 2016, vol. 4, p. e2364.
- [3] Kulic, D., Venture, G. et Nakamura, Y. (2009). "Detecting changes in motion characteristics during sports training". *IEEE Engineering in Medicine and Biology Society*, volume 2009, p. 4011–4.
- [4] Gagey, O., Bonfait, H., Gillot, C., & Mazas, F. (1988). "Anatomie fonctionnelle et mécanique de l'élévation du bras. *Revue de chirurgie orthopédique et réparatrice de l'appareil moteur*", 74(3), 209-217.
- [5] B. Loann. "Les mouvements en mécanique classique" [En ligne]. 2015. Disponible sur : <http://physiquereussite.fr/les-mouvements-en-mecanique-classique> [consultée le 04/01/2020].
- [6] Yann. "Décrire le mouvement d'un point" [En ligne]. 2017. Disponible sur : <https://www.superprof.fr/ressources/scolaire/physique-chimie/seconde/mecanique/decrire-un-mouvement.html> [consultée le 04/01/2020].
- [7] G. Outrequin, B. Boutillier. "Introduction aux chaînes musculaires cinétiques (CMC) " [En ligne]. 2016. Disponible sur : <https://www.anatomie-humaine.com/Notions-generales.html> [consultée le 05/01/2020].
- [8] Keith L. Moore, Arthur F. Dalley, 1999 , "Clinically Oriented Anatomy". Wolter Kluwer 2017
- [9] J. Hamilton. "Kinematics of Human Motion " [En ligne]. 2017. Disponible sur : <https://study.com/academy/lesson/kinematics-of-human-motion.html> [consultée le 06/01/2020].
- [10] P. Sylvestre [En ligne]. 2015. Disponible sur : https://www.cap-concours.fr/enseignement/preparer-les-concours/les-epreuves-du-crpe/les-divers-mouvements-du-corps-humain-mas_vie_17. [consultée le 15/01/2020].

- [11] Alain Viala, “ Posture”, dans Anthony Glinoe et Denis Saint-Amand (dir.), Le lexique socius, URL : <http://ressources-socius.info/index.php/lexique/21-lexique/69-posture>, page [consultée le 08/01/2020] .
- [12] Viala (Alain), “ Eléments de sociopoétique”, dans Approches de la réception. Sémiostylistique et sociopoétique de Le Clézio, sous la direction de Georges Molinié & Alain Viala, Paris, Presses Universitaires de France, « Perspectives littéraires », 1993, p. 216.
- [13] A. Viala, “ Naissance de l’écrivain”. Sociologie de la littérature à l’âge classique, Paris, Minuit, 1985. Revue française de sociologie. 27-3 pp. 584-586.
- [14] Massion J. “Postural control system”. Curr Opin Neurobiol 1994 ; 4 : 877–87.
- [15] Winter D. Human balance and posture control during standing and walking. Gait Posture 1995 ; 3 : 193–214.
- [16] Z. Cheng, Kathleen Robinette et INFOSCITEX CORP DAYTON OH . “ Static and Dynamic Human Shape Modeling - A Review of the Literature and State of the Art”. Defense Technical Information Center, 2009.
- [17] (Anne Claire N ,2017) , “Posture et équilibre : deux notions différentes”. Disponible sur : <https://www.elsevier.com/fr-fr/connect/kine-osteo/posture-et-equilibre-deux-notions-differentes>. [consultée le 19/01/2020].
- [18] Chris HAYOT “Analyse biomécanique 3D de la marche humaine : Comparaison des modèles mécaniques”, 2010.
- [19] MÉNARD, David. “Génération d'animations par capture de mouvements avec multiples caméras 3D”. 2014. Thèse de doctorat. École Polytechnique de Montréal.
- [20] “Capture de mouvement” [En ligne]. Disponible sur : <http://gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx> [consultée le 15/01/2020].
- [21] GIANNI, Frédérick. “Suivi de parties de corps pour l'interprétation de gestes de communication à partir de séquence monoculaire”. 2007. Thèse de doctorat. Université de Toulouse, Université Toulouse III-Paul Sabatier
- [22] “Capture de mouvement. 2020”. wikiwand, [En ligne]. Disponible sur : http://fr.wikipedia.org/w/index.php?title=Capture_de_mouvement&oldid=174103508 [Consultée le 30/01/2020]

- [23] SHOTTON, Jamie, SHARP, Toby, KIPMAN, Alex, et al. “Real-time human pose recognition in parts from single depth images”. *Communications of the ACM*, 2013, vol. 56, no 1, p. 116-124.
- [24] Sun Joo Ahn, Jesse Fox et Jeremy N Bailenson. “Avatars” In: *Leadership in science and technology: A reference handbook* (2012), p. 695–702.
- [25] Jeremy N Bailenson et Jim Blascovich. “Avatars”. In: *Encyclopedia of human-computer interaction*, Berkshire Publishing Group. Citeseer. 2004.
- [26] Jean Segura. “Une application qui crée des avatars impressionnants pour la VR”. [En ligne]. 2020. Disponible sur : <https://www.realite-virtuelle.com/definition-realite-virtuelle/> [Consulté le 02/02/2020].
- [27] Jantakoon, Thada & Wannapiroon, Panita & Nilsook, Prachyanun. (2019). “Virtual Immersive Learning Environments (VILEs) Based on Digital Storytelling to Enhance Deeper Learning for Undergraduate Students ”. *Higher Education Studies*. 9. 10.5539/hes.v9n1p144.
- [28] OQBI, Djedda. “Animation d’un humanoïde virtuel par une méthode de capture de Mouvement”. 2016. Thèse de doctorat. Université Mohamed Khider-Biskra.
- [29] “Nos techniques d’animation 3D–studio 3D emotion”. [En ligne]. 2016. Disponible sur : <https://www.3demotion.net/nos-techniques-danimation-3d/>. [Consulté le 09/04/2020].
- [30] FOLEY, James D., VAN DAM, Andries, HUGHES, John F., et al. *Spatial-partitioning representations; Surface detail.* ” *Computer Graphics: Principles and Practice*, 1990.
- [31] CHANDRU, Vijay, MAHESH, N., MANIVANNAN, M., et al. “Volume sculpting and keyframe animation system”. In : *Proceedings Computer Animation 2000*. IEEE, 2000. p. 134-139.
- [32] REULEAUX, Franz. “The Kinematics of Machinery”, (trans. and annotated by ABW Kennedy). 1963.
- [33] “MODELISATION CINEMATIQUE DES CHAINES DE SOLIDES” - PDF. 2016. Docplayer.fr [en ligne]. Disponible sur : <https://docplayer.fr/16968607-Modelisationcinematique-des-chaines-de-solides.html>. [Consulté le 10/04/2019].

- [34] “L’animation par cinématique directe ou inverse” . Slideplayer.fr [en ligne]. 2018. Disponible sur : <https://slideplayer.fr/slide/1534541/>. [Consulté le 12/04/2020].
- [35] WATT, A. et WATT, M. “Advanced Animation and Rendering Techniques: Theory and Practice”, Addison. 1992.
- [36] Olivier Vaillancourt, Olivier Godin. “CINÉMATIQUE DIRECTE ET INVERSE”. Cours en ligne. Canada. Université de Sherbrooke. Disponible sur : info.usherbrooke.ca/ogodin/enseignement/imn538/chapitres/imn538-chap06.pdf. [Consulté le 13/04/2020].
- [37] Nicolas Holzschuch. “Cinématique Inverse, Cours d’Option Majeure 2” [En ligne]. Disponible sur : http://maverick.inria.fr/Members/Nicolas.Holzschuch/cours/class4_small.pdf. [Consulté le 14/04/2020].
- [38] Jean-Thomas Masse. “Capture de mouvements humains par capteurs RGB-D”. Robotique. Thèse. Université Paul Sabatier - Toulouse III, 2015. France.
- [39] M. Bonenfant. “Capture De Mouvement Par Fusion De Poses Multivues Pour Une Réadaptation En Environnement Virtuel”. 2017. Mémoire Université Laval Québec, Canada.
- [40] LUN, Roanna et ZHAO, Wenbing. “A survey of applications and human motion recognition with microsoft kinect.” In”ternational Journal of Pattern Recognition and Artificial Intelligence, 2015, vol. 29, no 05, p. 1555008.
- [41] HOIEM, Derek. “How the kinect works. Lecture notes for Computational Photography”, 2011
- [42] Bertrand, PECUCHET . “Kinect et la reconstruction 3D”.2013. Rapport universitaire Licence. Université de lille 1.
- [43] KHOSHELHAM, Kourosh. “Accuracy analysis of kinect depth data”. In : ISPRS workshop laser scanning. 2011
- [44] ANDRÉ, Cyrille. “*Approche crédibiliste pour la fusion multi capteurs décentralisée*”. 2013. Thèse de doctorat. Paris 11.

- [45] David Bellot. “Fusion de données avec des réseaux bayésiens pour la modélisation des systèmes dynamiques et son application en télémédecine”. Autre [cs.OH]. Université Henri Poincaré - Nancy 1, 2002.
- [46] Khaleghi, Bahador & Khamis, Alaa & Karray, Fakhri & Razavi, Saiedeh. (2013). “Multisensor Data Fusion: A Review of the State-of-the-art. Information Fusion - INFFUS. 14”. 10.1016/j.inffus.2011.08.001.
- [47] DURRANT-WHYTE, Hugh et HENDERSON, Thomas C. “Multisensor data fusion”. Springer handbook of robotics, 2008, p. 585-610.
- [48] Al Hage, Joelle. (2016). “Fusion de données tolérante aux défaillances : application à la surveillance de l’intégrité d’un système de localisation”.
- [49] MAYBECK, Peter S. “The Kalman filter: An introduction to concepts”. In : Autonomous robot vehicles”. Springer, New York, NY, 1990. p. 194-204.
- [50] DOUCT, A., DE FREITAS, S. J. G., et GORDON, N. “Sequential Monte Carlo in Practice”. New York: Springer-Verlag, 2001, 2001, vol. 25, p. 1540-1550.
- [51] J. Rechar. Rehabilitation & Human Performance. 2020. qualisys.com [en ligne]. Disponible sur: <https://www.qualisys.com/stories/gait-rehab-salford> [consulté le 02/04/2019].
- [52] P. Gary. “Motion Capture Technology To Be Used To Train Sporting Hopefuls”. 2014. factor-tech.com [en ligne]. Disponible sur: <https://www.factor-tech.com/3d-modelling/8647-motion-capture-technology-to-be-used-to-train-sporting-hopefuls> [consulté le 02/04/2020].
- [53] A. Kitsikidis et al. “Dance Analysis using Multiple Kinect Sensors”. DOI: 10.5220/0004874007890795. In Proceedings of the 9th International Conference on Computer Vision Theory and Applications (IAMICH-2014), pages 789-795
- [54] R. Nick. “How’d They Do That ? A Brief Visual History of Motion-Capture Performance on Film”. screencrush.com [en ligne]. Disponible sur: <https://screencrush.com/motion-capture-movies> [consulté le 02/04/2020].
- [55] H. Vincent. “Kinect déjà détourné : 2000 dollars de récompense pour un pilote”. 2010 nextinpact.com [en ligne]. Disponible sur: <https://www.nextinpact.com/archive/60272-kinect-concours-hacking-utilisation-xbox-360-detournement.htm> [consulté le 06/04/2020].

- [56] K. Berger et al., “Markerless Motion Capture Using Multiple Color-Depth Sensors”, Vision, Modeling, and Visualization, Airela Ville, Switzerland: The Eurographics Association, 2011, pp. 317–324.
- [57] L. Zhang et al., “Real-Time Human Motion Tracking Using Multiple Depth Cameras,” IEEE Int. Conf. Intell. Robots Syst., Vilamoura-Algarve, Portugal, Oct. 7–12, 2012, pp. 2389–2395.
- [58] B. Williamson et al., “Multi-kinect Tracking for Dismounted Soldier Training”, Interservice/Ind. Training, Simulation, Educ. Conf., Orlando, FL, USA, Dec. 3–6, 2012, pp. 1-9.
- [59] S. Kaenchan et al., “Automatic Multiple Kinect Cameras Setting for Simple Walking Posture Analysis,” Int. Comput. Sci. Eng. Conf., Bangkok, Thailand, Sept. 4–6, 2013, pp. 245–249.
- [60] A. Kitsikidis, K. Dimitropoulos, S. Douka, and N. Grammalidis, “Dance analysis using multiple Kinect sensors,” in Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), 2014.
- [61] S. Moon et al., “Multiple Kinect Sensor Fusion for Human Skeleton Tracking Using Kalman Filtering,” Int. J. Adv. Robot. Syst., vol. 13, 2016, pp. 1–10.
- [62] H. Jo et al., “Motion Tracking System for Multi-user with Multiple Kinects,” Int. J. u- e-Service, Sci. Technol., vol. 8, no. 7, 2015, pp. 99–108.
- [63] N. Ahmed, “Unified Skeletal Animation Reconstruction with Multiple Kinects”. Aire-la Ville, Switzerland: the Eurographics Association, 2014, pp. 5–8.
- [64] Y. Kim et al., “Motion Capture of the Human Body Using Multiple Depth Sensors”. ETRI Journal, Volume 39, Number 2, 2017, pp. 181–190.
- [66] OUSMER, Mehdi. 2017. “Motion capture pour l’animation et l’augmentation d’avatars”. Mémoire de Master. Informatique visuelle. Alger : Université des sciences et de la technologie Houari-Boumediene
- [67] MOLYANSKIY, Nikolai, HUITEMA, Christian, LIANG, Lin, et al. “Real-time 3D face tracking based on active appearance model constrained by depth data”. Image and Vision Computing, 2014, vol. 32, no 11, p. 860-869.