

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOHAMED KHIDER DE BISKRA

Faculté des sciences exactes et sciences de la nature et de la vie

Département d'informatique

N° d'ordre:.....

N° de Série:.....



Mémoire

Présenter en vue de l'obtention du diplôme de

Magister en Informatique

Option :Data Mining et Multimédia

Thème

Animation d'un humanoïde virtuel par une méthode de capture de Mouvement

Présentée par : DJEDDA Oqbi

Soutenu le : 03/ 07 / 2016

Devant le jury:

Président : MELKEMI Kamal Eddine	Maître de conférences 'A'	Université de Biskra.
Rapporteur : BABAHENINI Med Chaouki	Maître de conférences 'A'	Université de Biskra.
Examineur: BITAM Salim	Maître de conférences 'A'	Université de Biskra.
Examineur: DJEROU Leila	Maître de conférences 'A'	Université de Biskra.

Remerciement

Je remercie "Dieu" de nous avoir donné la patience, la santé, le courage d'avoir arrivé jusque là.

*Je tiens à remercier le docteur **Mohamed Chaouki BABAHENINI** qui m'a honoré par son encadrement, pour sa direction, son orientation, ses conseils et toutes ses remarques constructives pour le bon déroulement de notre travail.*

*Je tiens à remercier Dr. **Kamel Eddine MELKEMI**, Dr. **Salim BITAM** et Dr. **Leila DJEROU** d'avoir accepté d'être les examinateurs.*

*Merci à mes parents de m'avoir amené jusqu'ici, merci également à toute ma grande famille, frères, sœurs surtout **Amina** et **Wafa**, neveux et nièces.*

*Enfin, je remercie **ma femme** qui m'a encouragé et qui a su me supporter, pour le meilleur et pour le pire pendant ces années.*

*A ma fille **Amna** et mes fils **Med El Amine** et **Amar**.*

Enfin, nous remercions, avec toute la suprême sincérité, tous ceux qui nous ont aidés de près ou de loin à terminer et à réaliser ce travail.

OQBI DJEDDA

ملخص

هناك عدة طرق لمحاكاة حركة البشر في العالم الافتراضي، ومن بين أهم هذه الطرق تحرير الحركة، التي تسمح بالحصول على حركة جديدة استناداً إلى حركة موجودة من قبل في قاعدة الحركات.

وهناك العديد من الشركات التي طورت ملفات الالتقاط الحركات (قاعدة الحركات)، مثل بيوفزون . بيو ميكانيك وموشن أناليز. وأصبح لدينا قاعدة ضخمة.

عند تحرير الحركة و التعديل يجب الحفاظ على بعض خصائص الحركة وطبيعتها الأصلية للحصول على حركة واقعية و متناسقة.

كجزء من هذه المذكرة، نود أن نقدم نموذج لتحرير الحركة البشرية. لذلك اقترحنا نموذج لتحرير الحركة يعتمد على الكينماتيكا العكسية (الحركة العكسية). في هذا الصدد نقترح نموذجين لتحرير الحركة، نستخدم في أول نموذج في حركة الإنسان الافتراضي الحركة الملتقطة من الواقع. و قمنا في النموذج الثاني باستخدام الحركة الملتقطة من الواقع و الكينماتيكا العكسية التي تعتمد علي طريقة فابريك. والإنسان الافتراضي هو نموذج ثلاثي الأبعاد.

اهتمنا بتطوير الجوانب التالية: إدماج وتكييف الإنسان الافتراضي مع الحركات الملتقطة من الواقع. دمج طريقة فابريك على الإنسان الافتراضي أثناء الحركة.

كلمات البحث:

تحرير الحركة، الإنسان الافتراضي، الحركة الملتقطة من الواقع، طريقة فابريك.

Résumé

Il existe plusieurs méthodes pour la simulation du mouvement des humains dans un monde virtuel, parmi les plus importantes l'édition de mouvement, qui permet d'obtenir un nouveau mouvement en se basant sur un mouvement qui existe déjà (base de mouvement), pour cela plusieurs sociétés comme la société Biovision, Biomechanics et Motion Analysis ont développé des fichiers de capture de mouvements (base de mouvement).

Lors de l'édition ou la modification du mouvement, on a besoin de modifier les données sans détruire le mouvement naturel et préserver quelques caractéristiques du mouvement original afin d'obtenir un mouvement réaliste et contenu.

Dans le cadre de ce mémoire de magister, nous souhaitons apporter des contributions à l'édition mouvement. Nous proposons une technique d'édition de mouvement basée sur la cinématique inverse. Nous proposons deux modèles d'édition de mouvement, dans le premier modèle l'animation de l'humanoïde virtuel est modélisé uniquement par la capture de mouvement, et dans le deuxième modèle l'animation est réalisée par la capture de mouvement et la cinématique inverse basé sur la méthode FABRIK. L'humanoïde virtuel est modélisé par un modèle 3D.

Nous nous sommes intéressés à développer les aspects suivants: l'intégration et l'adaptation des mouvements capturés à nos humanoïdes, et l'utilisation de la méthode FABRIK sur l'humanoïde en animation.

Mots clés : Edition de mouvement, humanoïde virtuel, capture de mouvement, méthode FABRIK.

Abstract

There are several methods for the simulation of the movement of humans in a virtual world, among the most important of these methods editing motion, which allows to obtain a new movement based on a movement already exist (base movement). Which allows to obtain a new movement based on a movement already exist (base movement). There are different company that have developed files motion capture (motion base) as the company Biovision, Biomechanics and Motion Analysis. And we have become an important base.

When editing motion (motion to change), one must need to change the data without destroying the natural movement and preserve some features of the original movement to obtain a realistic and content movement.

As part of this thesis, we want to provide contributions to publishing movement. We offer a publishing movement technique based on inverse kinematics. We offer two models of publishing movement in the first model the virtual humanoid animation is modeled by motion capture only, and the second model animation and motion capture inverse kinematics based on FABRIK method. The virtual humanoid is modeled by a 3D model.

We were interested to develop the following: integration and adaptation of our movements captured humanoids, integrate FABRIK method on the humanoid animation.

Keywords: editing motion, virtual humanoid, motion capture, FABRIK method.

Table des matières

I. Introduction générale	1
II. Chapitre 1 : Humanoïde virtuel et génération de mouvement.....	3
1.1 Introduction	4
1.2 Humanoïde virtuel	4
1.3 Modélisation des humanoïde virtuel	5
1.4 Degré de liberté (DDL)	9
1.5 Technique de génération des mouvements	10
1.5.1 Par images clé	10
1.5.2 Animation procédurale	11
1.5.3 Simulation physique.....	12
1.5.4 Animation basée sur la capture de mouvement	13
1.6 Conclusion.....	16
III. Chapitre 2 : Capture de mouvements et Edition de mouvements	17
2.1 Introduction	18
2.2 Capture de mouvement.....	19
2.2.1 Domaine d'application	19
2.2.2 Les systèmes de capture du mouvement.....	20
2.2.2.1 La capture mécanique.....	20
2.2.2.2 La capture magnétique	22
2.2.2.3 Système optique	23
2.2.3 Processus de capture du mouvement	25
2.2.4 Fichiers des mouvements capturés	27
2.2.4.1 Fichiers BVH (Biovision Hierarchical Data).....	27
a) Structure du fichier BVH	27
b) La partie hiérarchique :	28
c) La partie mouvement.....	29
2.3 Edition de mouvement.....	30
2.3.1 Méthodes d'édition de mouvement	30
a) Mélange de mouvements « Motion Blending »	31
b) Spatio-temporelle «Space time Constraints »	32
c) Adaptation de mouvements « Motion Retargeting »	33
d) Déformation cinématique « Motion warping ».....	34
e) Cinématique inverse «Inverse kinematics »	35
2.4 Conclusion.....	37

IV. Chapitre 3: Technique d'édition de mouvement basée sur la cinématique inverse	38
3.1 Introduction	39
3.2 Cinématique directe.....	40
3.3 Cinématique inverse	40
3.4 Méthodes de résolutions de la cinématique inverse	41
3.4.1 Méthodes analytiques.....	41
3.4.2 Méthodes itératives	42
a.1) Méthodes basées sur la Jacobienne	42
a.2) La Jacobienne transposée.....	43
a.3) La pseudo-inverse	43
3.4.3 Méthode basée sur le gradient	44
a.1) Descente de gradient	44
a.2) Méthode Monte-Carlo.....	45
3.4.4 Méthode Descente de Coordonnées Cycliques CCD.....	45
3.4.5 Méthode FABRIK (Forward And Backward Reaching Inverse Kinematics)	46
3.4.6 Bilan.....	47
3.5 Conclusion.....	48
V. Chapitre 4: Animation d'un humanoïde virtuel par la méthode FABRIK	49
4.1 Introduction	50
4.2 Méthode FABRIK (Forward And Backward Reaching Inverse Kinematics)	51
a) Principe de fonctionnement.....	51
b) Itération complète de FABRIK (exemple)	54
c) FABRIK avec contraintes	56
d) Contrainte sur la jointure	56
e) Algorithme pour les restrictions d'orientation.....	57
f) Algorithmes pour les restrictions de rotation	58
4.3 Technique d'animation d'un humanoïde virtuel par FABRIK.....	62
4.3.1 Module 1 : animation par capture des mouvements	64
a) Lecture le fichier BVH.....	64
a.1) Structures des données	64
b) Relier le squelette de BVH avec le squelette de l'Humanoïde (mapping) ou Plaquage des mouvements	66
c) Utilisation de la matrice BVH pour animer l'humanoïde	67
4.3.2 Module 2 : utilisé la cinématique inverse afin d'atteindre une cible (intégré la méthode FABRIK sur l'humanoïde).....	69
a) Modèle humanoïde.....	69

a.1) Définition les jointures utilisées pour l'animation :	70
a.2) Les chaines et Les effecteurs de fin	71
b) Edition de mouvement basée sur la cinématique inverse (FABRIK)	72
b.1) Extraction des positions des jointures de l'humanoïde :	72
b.2) Appliquer FABRIK sur le corps de l'humanoïde :	73
c) Intégration de l'algorithme FABRIK, pour le retour de l'humanoïde à la pose de frame suivante	75
4.4 Conclusion.....	76
VI. Chapitre 5 : Mise en œuvre, résultats et bilan	77
5.1 Introduction	78
5.2 Environnement de développement.....	78
5.2.1 Présentation de Microsoft VisualStudio 2010 Ultimate :	78
5.2.2 3DS MAX	79
5.2.3 Ogre 3D.....	79
5.2.3.1 L'entité	80
5.2.3.2 Le SceneManager.....	81
5.2.3.3 Ogre Exporter.....	81
5.2.4 Représentation de l'interface développée.....	81
5.3 Utilisation d'un mouvement capturé pour animer l'humanoïde :	83
5.3.1 Mise en correspondance entre les jointures de fichier du mouvement et le personnage virtuel.....	83
5.3.2 Résultats	84
5.3.2.1 Adaptation du mouvement de la marche	84
5.3.2.2 Adaptation du mouvement de la course.....	85
5.3.2.3 Adaptation du mouvement de la boxe	85
5.3.2.4 Adaptation du mouvement de la boxe sur deux différents personnages virtuels	86
5.4 Applique (intégré) FABRIK pour modifier le mouvement de base (édition).	88
5.4.1 Résultats	88
5.4.1.1 Adaptation du mouvement de la marche pour atteint un objet	88
5.5 Conclusion.....	93
Conclusion générale.....	94
Bibliographie	97

Liste des figures

Chapitre I : Humanoïde virtuel et génération de mouvement

Figure 1.1: Représentation sous forme d'arbre.....	05
Figure 1.2: Représentation d'un squelette humain.....	07
Figure 1.3: Les six degrés de liberté.....	10
Figure 1.4 : Interpolation entre deux dessins avec personnage d'interpolation.....	11
Figure 1.5: Exemple des méthodes physiques.....	13
Figure 1.6 : Exemple des méthodes physiques.....	14

Chapitre 02 : Capture de mouvements et Edition de mouvements

Figure 2.1: Animation de personnages guidés par des données capturées dans les films d'animation et les jeux.....	20
Figure 2.2: System mécanique GypsyGyro (société Gypsy).....	21
Figure 2.3: Exemple de la capture magnétique.....	22
Figure 2.4: Exemple de marqueurs actifs.....	24
Figure 2.5 : Exemple de marqueurs passifs.....	25
Figure 2.6 : La structure du squelette d'un fichier d'exemple BVH.....	28
Figure 2.7 : Exemple de la partie hiérarchique du fichier BVH.....	29
Figure 2.8 : Exemple de la partie mouvement du fichier BVH.....	30
Figure 2.9 : Mélange de mouvements.....	31
Figure 2.10: Deux mouvements des marches avec temps différentes.....	32
Figure 2.11 : Déformation cinématique.....	35
Figure 2.12 : Exemples de contrôle de chaîne articulée par cinématique inverse.....	35

Chapitre 03 : Réalisation d'une technique d'édition de mouvement basée sur la cinématique inverse

Figure 3.1: Principe de la cinématique inverse.....	41
Figure 3.2: Les angles des articulations permettant d'obtenir une telle configuration.....	42
Figure 3.3: Exemple d'utilisation du CCD.....	45
Figure 3.4: figure présente la seconde itération qui suit celle de la figure 3.3.....	46

Chapitre 04 : Animation d'un humanoïde virtuel par la méthode FABRIK

Figure 4.1: Itération complète de FABRIK.....	54
Figure 4.2 : Chaîne avec plusieurs sous-bases	55
Figure 4.3 : Cône représentant Les restrictions de rotations	56
Figure 4.4 : Différentes sortes de cônes	57
Figure 4.5 : Contrainte d'orientation.....	58
Figure 4.6 : Les possibilités de positionnement pour t.....	60
Figure 4.7 : FABRIK avec des contraintes d'orientation et de rotation	61
Figure 4.8 : Schéma proposé pour gérer l'application	62
Figure 4.9 : Schéma Détail proposée	63
Figure 4.10 : Lecture fichier BVH.....	64
Figure 4.11 : Modèle humanoïde : hiérarchie proposée	69
Figure 4.12 : Modèle humanoïde : degré de liberté.....	70
Figure 4.13 : Les chaînes et Les effecteurs de fin	71
Figure 4.14 : Appliquée FABRIK entre les frames, pour éditer nouvelle mouvement.....	72

Chapitre 05 : Mise en œuvre, résultats et bilan

Figure 5.1 : Modèle de l'humanoïde 3D	79
Figure 5.2: L'interface de première version d'application	82
Figure 5.3: L'interface de deuxième version d'application	82
Figure 5.4: Information sur l'état de la simulation	83
Figure 5.5 : Fichier de mappage squelette	84
Figure 5.6: Adaptation du mouvement de la marche.....	85
Figure 5.7: Adaptation du mouvement de la course	85
Figure 5.8: Appliqué le mouvement de la boîte sur notre personnage virtuel.....	86
Figure 5.9 : Application du même mouvement de la boîte sur les deux personnes.....	87
Figure 5.10: FARIBK utilise un seul chaîne pour atteint un objet.....	89
Figure 5.11: FARIBK utilise deux chaîne pour atteint un objet.....	89
Figure 5.12: FARIBK utilise deux chaîne pour atteint un objet, gardez les pieds sur terre.....	90

Figure 5.13: Atteindre un objet seulement avec le pied	90
Figure 5.14: Les pieds de personnage sont sur le sol et atteint l'objet	91
Figure 5.15: Eviter la collision entre le pied et la table	91
Figure 5.16: Graphe de comparaison entre posture capturé et une autre construite par la méthode proposée	93
Figure 5.17: Graphe du temps de calcul mesuré par rapport au nombre de triangles entre posture capturé et une autre construite par la méthode proposée.....	93

I. Introduction générale

La simulation et l'analyse du mouvement humain intéressent de nombreux domaines d'application. . Dans la robotique, l'utilisation d'humanoïde est également un domaine qui s'inspire de l'étude du mouvement humain pour animer des robots. Dans le sport, l'étude du mouvement et du geste a pris une grande importance pour améliorer les performances des sportifs. Dans le domaine médical, on cherche à comprendre et à interpréter au mieux le mouvement humain dans le cas des personnes handicapées. Dans les domaines du multimédia (jeux vidéos, cinéma, films), elle a pour objectif d'animer des humains virtuels de manière réaliste.

Il existe des nombreuses méthodes qui nous permettent de simuler un mouvement humain : soit en utilisant des mouvements réels extraits d'une base des mouvements (base des capture des mouvements) et les adapter à un humanoïde de synthèse ou extraire leur caractéristiques et les combiner avec un autre mouvement afin d'obtenir un autre mouvement complexe. Soit en créant le mouvement en utilisant une méthode par images clés.

L'édition de mouvement est une technique qui permet d'obtenir un nouveau mouvement en se basant sur un mouvement qui existe déjà dans une base de mouvement. Il existe plusieurs méthodes d'édition de mouvement tel que : motion warping, motion blending ...etc, lors de la modification de mouvement, on doit préserver quelques caractéristiques du mouvement original afin d'obtenir un mouvement réaliste et contenu.

Problématique :

L'objectif de notre travail consiste à modifier un mouvement qui existe déjà (sur la base du mouvement) en respectant la continuité de mouvement résultat et en choisissant un intervalle de temps où on doit modifier le mouvement ainsi que la définition des caractéristiques à préserver. Pour atteindre l'objectif de notre projet ; nous nous proposons de traiter les points suivants :

1. Modélisation d'un humanoïde virtuel en utilisant un logiciel graphique,
2. Modélisation d'un environnement virtuel (chambre avec des murs et une table).
3. Utilisation d'un mouvement capturé (définition du contenu de fichier de mouvement, adaptation de mouvement aux différentes morphologies) pour animer l'humanoïde virtuel.

4. Proposer une approche d'édition de mouvement qui permet de modifier le mouvement de base (Modification du mouvement).

Organisation du mémoire

Pour cela, nous avons entrepris notre étude selon les quatre chapitres suivants :

Dans le premier chapitre, nous étudierons quelques notions sur l'Humanoïde Virtuel (définition, modélisation, caractéristique...etc) et la génération de mouvement (types, avantages et inconvénients chacun d'entre eux).

Le deuxième chapitre, est consacré d'une part à une étude sur la capture de mouvements, où nous allons exposer les domaines d'application, les différents types de systèmes de capture du mouvement, les processus de capture du mouvement et le contenu des fichiers de mouvements capturés et des fichiers BVH et d'autre part à l'édition de mouvements où nous allons détailler principalement les méthodes d'édition de mouvement.

Dans le troisième chapitre, nous présentons notre technique de modification du mouvement à base de la cinématique inverse, son principe de base consiste à découper le personnage virtuel en groupes, et ensuite à appliquer la cinématique la méthode FABRIK (Forward Backward Inverse Kinematic), qui parcourt des chaînes en avant et en arrière afin d'atteindre l'objectif, l'application de FABRIK sera réalisé pour chaque groupe tout en prenant compte des relations entre les groupes.

Le quatrième chapitre, présente la partie implémentation et validation de nos résultats.

Une conclusion des travaux présentés dans ce mémoire ainsi que des perspectives de leur utilisation seront présentées dans le dernier chapitre.

II. Chapitre 1 : Humanoïde virtuel et génération de mouvement

1.1 Introduction

Notre objectif est de couvrir toute les choses reliées par la simulation d'un mouvement humain. Ce chapitre est composé de deux parties :

La première partie définit l'humanoïde virtuel ou personnage virtuel et les types des modélisation, nous commençons par le modèle filaire ensuite, nous décrivons les primitives volumiques et polygonales et nous expliquons le modèle par couches qui contient le squelette et l'enveloppe, où le personnage virtuel est fréquemment représenté comme une structure d'arbre avec un grand nombre de degrés de liberté (DDL).

Dans la deuxième partie, nous présentons les techniques de génération de mouvements, parmi les techniques les plus utilisées nous citons les techniques : par image clé, procédurale, physique et par capture de mouvement.

1.2 Humanoïde virtuel

Un humanoïde virtuel ou personnage virtuel est un robot dont l'apparence générale rappelle celle d'un corps humain. Généralement, les humanoïdes virtuels ont un torse avec une tête, deux bras et deux jambes, bien que certains modèles ne représentent qu'une partie du corps, par exemple à partir de la taille. Certains humanoïdes virtuels peuvent avoir un « visage », avec des « yeux » et une « bouche » [58].

Donc l'humanoïde c'est la représentation (reproduction) graphique par ordinateur d'un être humain dans un environnement virtuel [3].

Le corps humain est complexe avec environ 200 os, 600 muscles et 200 degrés de liberté, lorsqu'est entièrement modélisé avec des membres reliés rigides.

La nature déformable des parties du corps augmente encore cette complexité. Il faut donc trouver une représentation simplifiée de l'humanoïde afin de diminuer la complexité et simplification de l'être humain [2].

1.3 Modélisation des humanoïde virtuel

La modélisation est une création de personnages dans un environnement virtuel en 3D et intégration des données de ces personnages pour l'animation, il existe trois modèles définies comme suit :

1.3.1 Modélisation filaire

La représentation en fil de fer est la représentation basique pour la modélisation du corps humain, où les os sont représentés par des segments, et les articulations par des petites sphères, cette combinaison nous donne une structure articulée.

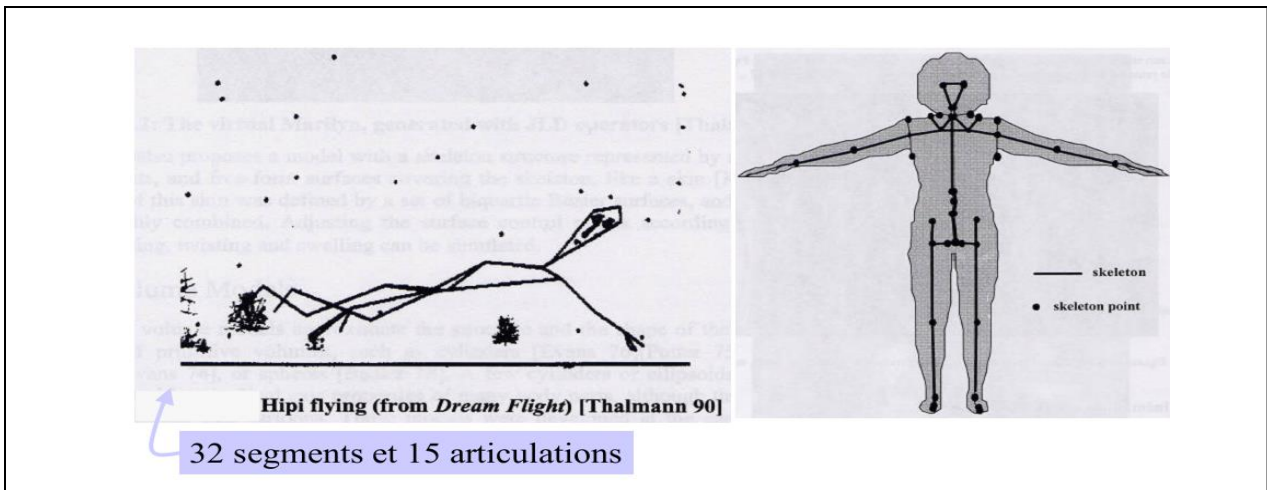


Figure 1.1 : Représentation sous forme d'arbre qui permet de traiter les différentes parties du corps de façon indépendante [11].

L'avantage principal de ce modèle est lié au fait que la spécification du mouvement est plus facile à générer; en associées simplement à chaque composante de l'arbre une matrice de transformation.

Néanmoins, ce modèle est limité par le manque de réalisme, perception de la profondeur est difficile et que certains mouvements sont impossibles à représenter.

1.3.2 Modélisation polygonale

Le modèle est assimilé à un ensemble de polygones souvent des triangles décrit par un ensemble de sommets (vertex) et des arrêtes ; cette modélisation est souvent appelée

modélisation triangulaire au maillage triangulaire (**Meshing**), la représentation du corps est créée manuellement à partir d'un des logiciels de modélisation courants comme (**MAYA, 3D Studio MAX, SOFTIMAGE 3D**).

Ce type de modélisation présente l'avantage de l'existence de l'application du modèle de rendu très efficace.

Par contre, il présente l'inconvénient lié au fait qu'il faut un nombre très important de polygones pour représenter une surface bien lisse.

1.3.3 Modélisation par primitives volumiques

La recherche d'un modèle toujours plus proche du modèle réel a conduit à considérer que l'humain virtuel est formé par des primitives volumiques simple tel que : des sphères pour les bras et les jambes, des ellipsoïdes pour la représentation de la tête, et même des cônes, sphères et des cubes platées. et à les combiner (généralement par des opérations booléennes)

Ceci permet d'avoir un modèle, qui à la capacité de se combiner avec des primitives voisines, malheureusement ce type de modélisation est très coûteux du point de vu calcul.

1.3.4 Modélisation par couches

Il s'agit de construire le personnage virtuel en plusieurs couches, dans le cadre de l'animation d'humanoïde, Chadwick et Coll [1]. ont proposé de décomposer le corps humain en trois couches distinctes : le squelette, les muscles et la peau.

1.3.4.1 Le squelette :

Le squelette humain est composé de plus 200 os [6], chaque os est relié à un ou plusieurs os par des liaisons (articulations). On peut représenter cette structure par un modèle hiérarchique, dont chaque segment est un os et chaque articulation représentée par un nœud qui relie deux os. Pour animer l'humanoïde de synthèse, il suffira de connaître la rotation de chaque segment du squelette (os). Le squelette est ainsi représenté par des solides rigides articulés.

La représentation du squelette par une représentation hiérarchique qui offre une formulation intuitive pour l'animation d'humanoïde de synthèse. Cependant, il ya quelques simplification qu'on peut appliquer, vu qu'il est difficile de contrôler avec précision l'ensemble

des os du squelette. L'objectif consiste donc à rendre le squelette humain moins complexe à manipuler. Donc il faut simplifier la structure par la manipulation d'un nombre d'articulation limité ce permet d'alléger la structure tout en conservant un niveau de détail suffisant pour retranscrire les principales caractéristiques d'un mouvement humain [59].

La diminution du nombre d'articulation implique donc le regroupement d'un sous ensemble d'os du squelette en une partie rigide. On peut citer comme exemple la colonne vertébrale qui à l'origine comporte 24 vertèbres et qui est généralement discrétisée en 5 ou 6 articulations afin de limiter la complexité du système. La figure 1.2 donne un aperçu d'une représentation du squelette humain couramment utilisée. La figure de gauche montre une représentation géométrique de l'ensemble des os composant le squelette humain, alors que celle du milieu propose un squelette simplifié par la limitation de nombre d'articulation. Enfin, la figure de droite montre un squelette simplifié sous une forme d'arborescente tel que la racine de squelette est le bassin.

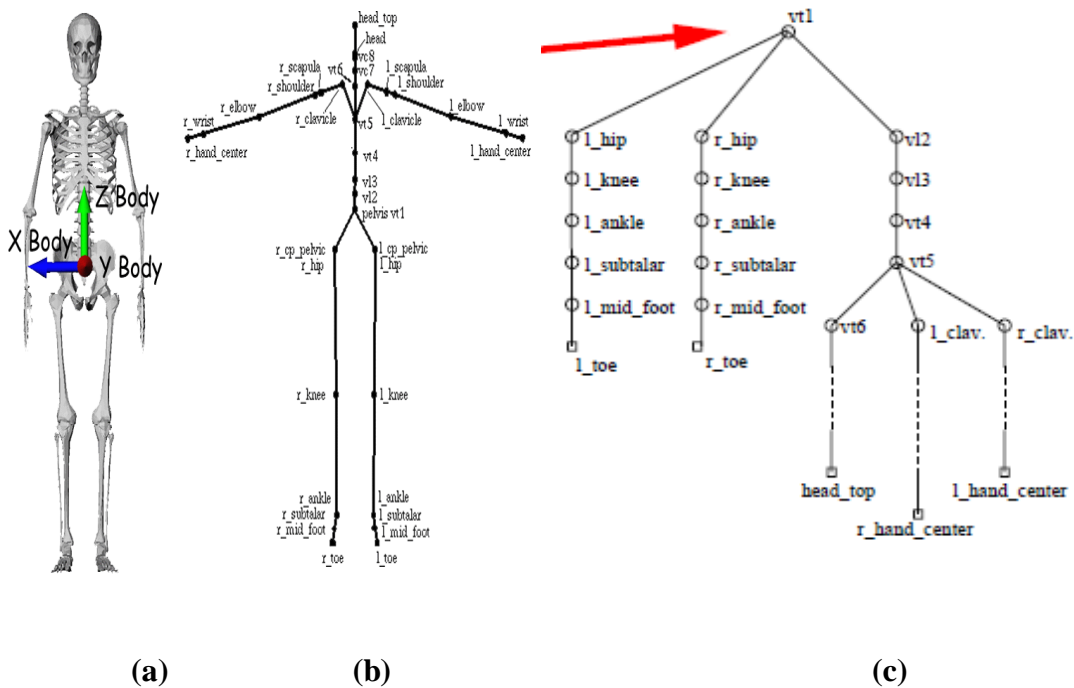


Figure 1.2 : **a :** représentation d'un squelette humain, **b :** squelette simplifié **c :** arborescence du squelette.

Une autre méthode de simplification du squelette consiste à simplifier la représentation des articulations, ces derniers sont associées à une position spatiale qui représente le centre de rotation entre deux objets rigides articulés. C'est la raison pour laquelle seuls les trois degrés de liberté de rotation sont utilisés pour déplacer les segments. Toutefois, sur un squelette humain, la zone de contact entre deux os ne permet pas d'avoir un unique centre de rotation, puisque celle-ci est surfacique et non parfaite. Plusieurs travaux se sont donc focalisés sur une représentation plus fidèle d'une partie complexe du squelette tel que la colonne vertébrale [20], l'épaule [8], la main [18] et le genou [19]. Ces études restent toutefois proposées pour des éléments séparés du corps humain. Ont focalise dans leurs travaux sur la partie de la colonne vertébrale du squelette, Mural et Thalmann [7] ont concentré sur l'épaule autre que Maciel [8] a focalisé sur la représentation de genou.

Une représentation du squelette articulé sous forme arborescente, suppose le choix d'une articulation particulière comme racine de l'arbre. On choisit généralement le centre du bassin comme centre (figure 1.2.c). Néanmoins, suivant le contexte de l'animation, ce choix n'est pas toujours bon. Ainsi, l'animation d'un humanoïde suspendu a une branche d'arbre par un doigt illustre clairement un squelette arborescent dont la racine devrait être associée à l'articulation de l'extrémité du doigt.

1.3.4.2 L'enveloppe (Les muscles et la peau)

Une fois l'arborescence établie, le graphiste doit rattacher au squelette une modélisation des muscles et de la peau pour définir une enveloppe corporelle. Les muscles s'appuient sur la structure squelettique afin de bouger les os. En effet, lorsqu'un muscle se contracte, sa longueur diminue de telle sorte que les os attaches se rapprochent. Ces muscles interviennent pour améliorer le réalisme de l'enveloppe corporelle de l'humanoïde synthétique, en changeant de forme en fonction de leur contraction.

La plupart des systèmes, actuellement mis en œuvre pour gérer un système musculaire sont coûteux et complexes à cause de la complexité musculaire du corps humains (presque 700 muscles). On peut regrouper les modèles des muscles en deux groupes : les modèles purement géométriques qui permettent de produire des animations temps réel et les modèles physiques à base de systèmes plus complexes.

- Modèle géométrique : [9] et la plupart des chercheurs ont représenté les muscles par un ellipsoïde, d'autres travaux ont utilisé les surfaces implicites [61] déduites par un ensemble d'ellipsoïdes pour représenter les muscles.
- Les modèles physiques, quant à eux, déforment l'enveloppe corporelle à l'aide de systèmes masse-ressort [10] ou par des techniques à base d'éléments finis.

La troisième couche concerne la couche de la peau des vêtements qui donne la touche finale au rendu du personnage et représente la surface visible du personnage. Cette couche subit des déformations provenant de la couche musculaire.

1.4 Degré de liberté (DDL) :

Le degré de liberté (DDL) ou ce qu'on l'appelle en anglais « Degree of freedom » (DOF) ou Channel représente les différentes possibilités de mouvement dans l'espace. Chaque os dans le squelette peut soumettre à une position et/ou orientation au cours de l'animation dont chaque paramètre est soumis à un degré de liberté (DDL).

Les changements dans les degrés de libertés au cours du temps donne l'évolution à l'animation [4]. La figure suivante représente différents DDL du corps.

Un corps possède six degrés de liberté, ce qui veut dire 6 mouvements sont considérés trois translations et trois rotations de directions indépendantes. Si on s'oriente dans l'espace (à 3 dimensions) à l'aide d'un repère orthonormé (O, x, y, z), les six degrés de liberté s'expriment comme suit :

a) En translation

- Avant - arrière (x)
- Droite- gauche (y)
- Haut - bas (z)

b) En rotation

- Basculer d'avant en arrière (Tangage)
- Basculer de droite à gauche (Roulis)
- Pivoter comme les aiguilles d'une montre (Lacet)

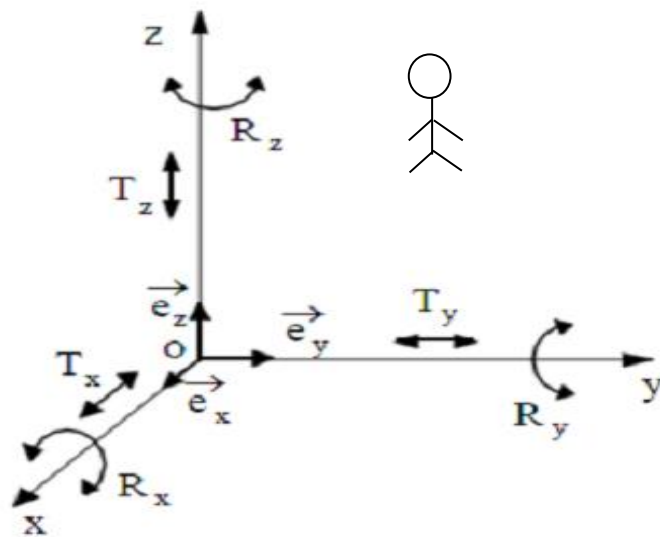


Figure 1.3: Les six degrés de liberté [5]

1.5 Technique de génération des mouvements

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

1.5.1 Par images clés

La plus simple des manières, pour modifier un mouvement consiste à modifier ce mouvement « à la main ». Cette méthode est héritée de l'animation 2D traditionnelle, Un animateur expérimenté dessine les principales images décrivant le mouvement, et d'autres dessinateurs complètent l'animation en dessinant les images intermédiaires.

L'ordinateur peut aussi se charger de calculer les images intermédiaires entre chaque couple d'images fournies en utilisant une méthode d'interpolation comme il est représenté dans la figure 1.4 [12].

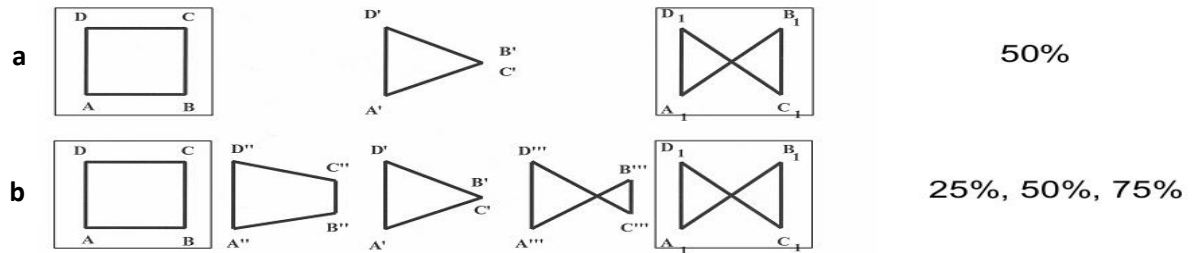


Figure 1.4 Interpolation entre deux dessins avec personnage d'interpolation.

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

- cette technique 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 :

- 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).

1.5.2 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 elle se base sur les algorithmes écrits pour la génération automatique des mouvements et souvent à l'aide des lois physiques.

Les méthodes procédurales s'appuient sur les algorithmes par main (handcrafted algorithms) pour générer automatiquement les mouvements. Ces algorithmes peuvent être considérés comme des fonctions variant au cours de temps pour contrôler les paramètres bas niveau du mouvement. Perlin [14] a appliqué les techniques de synthèse des textures procédurales afin de créer un humanoïde réactif en temps réel, l'auteur a réussi à générer un ensemble de mouvements variant. Ainsi Perlin et Goldberg [13] ont proposé « improv system » pour créer des humanoïdes de synthèse animés et réalistes. Chin et al [15] ont appliqué une

analyse de mouvement de Laban « Laban Motion Analysis » pour récupérer les aspects qualitatifs du mouvement, particulièrement ils se focalisent sur les propriétés de l'effort et de la forme. Donc, ces paramètres sont ajustés en temps réel pour modifier le sens et la mise en scène adoptée par l'acteur de synthèse. Boulic et al ont proposé un moteur de contrôle de la marche avec des paramètres de haut niveau comme le style, vitesse ou l'emplacement de sujet.

Les méthodes procédurales offrent l'usage de quelques types de systèmes de calcul pour améliorer la crédibilité de l'animation des humanoïdes de synthèse parce que l'humain virtuel peut se déplacer et réagir seul sans aucune force extérieure.

Généralement une procédure est dédiée à une classe d'animation. Donc, à chaque fois que l'animateur veut de nouveaux mouvements, il faut qu'il repense à un nouvel algorithme.

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.
- Lors qu'on a besoin des nouveaux mouvements, juste il suffit décrire un nouveau algorithme (procédure).

Mais elle est limitée par les inconvénients :

- 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).

1.5.3 Simulation physique

Les méthodes physiques ont presque le même principe de fonctionnement des méthodes procédurales. Néanmoins, au lieu de s'appuyer uniquement sur les fonctions de dessin par main, ces méthodes prennent en considération les lois physiques pour produire un nouveau mouvement. Elles sont appliquées avec succès pour l'animation des déformations des vêtements, les corps rigides et même pour générer les transitions entre mouvements comme la chute, le saut et les mouvements d'équilibre.

L'humain virtuel est d'abord considéré comme une structure à base des parties du corps pour lesquelles on connaît précisément la distribution des masses. Une distribution moyenne de

masses peut être facilement trouvée dans la littérature biomécanique. Alors, le problème est de trouver les couples d'articulations qui produisent le mouvement attendu.

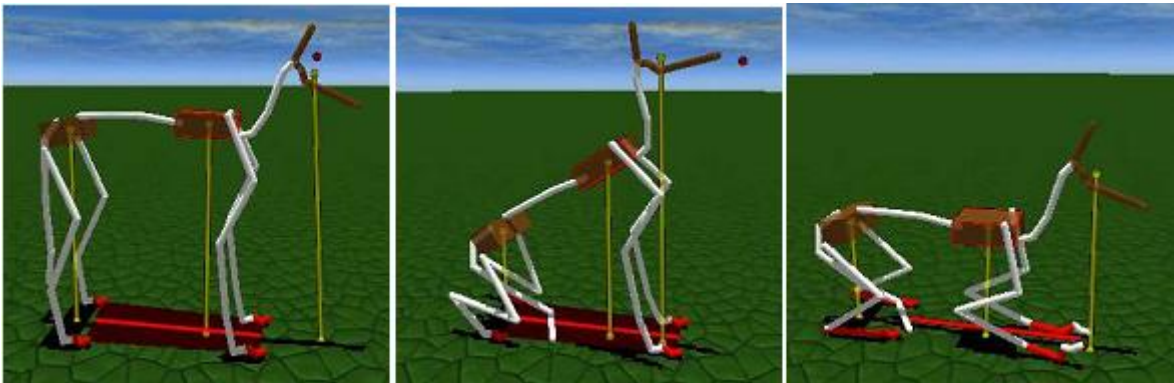


Figure 1.5 Exemple des méthodes physiques (chien Debout, assis et couché)

La simulation physique présente les avantages suivants :

- Beaucoup réaliste, car les lois sont inspirées de la nature.
- Pas de mouvements indésirables, mais exactes comme les lois.
- Capable de générer des mouvements physiquement corrects.

Mais elle est limitée par les inconvénients :

- Le temps de calcul est généralement très prohibitif, pour utiliser cette méthode dans les applications interactives.
- Nécessite une compréhension approfondie des paramètres physiques (masse, direction, gravité ...etc) afin d'obtenir les mouvements souhaités en sortie.

1.5.4 Animation basée sur la capture de mouvement

Dans cette méthode, on choisit une des techniques de capture de mouvement : optique avec ou sans marqueurs, magnétique ou mécanique pour traquer et enregistrer les mouvements effectués par l'acteur à l'aide des systèmes informatiques (matériels / logiciels) comme les marqueurs, les caméras, les récepteurs et les logiciels qui permettent de récupérer les positions de l'acteur, pour ensuite les exploiter, et les utiliser lors de la production d'image de synthèse. Ainsi on sera sûr, d'obtenir une animation réaliste [17].

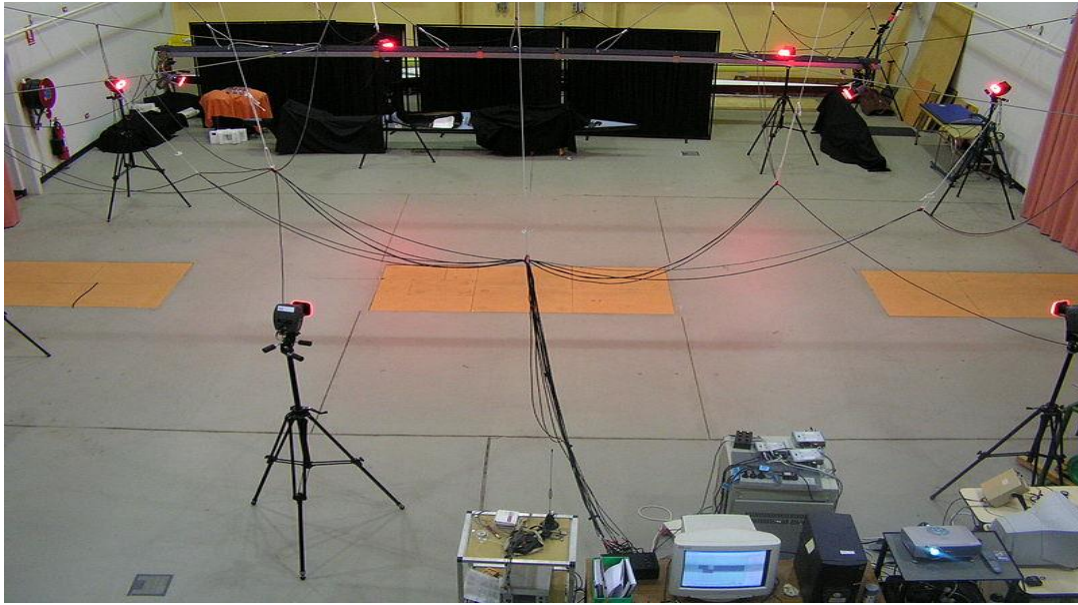


Figure 1.6 : Une salle d'enregistrement dédiée à la capture de mouvement, équipée de plateformes sensibles aux pressions et de caméras infrarouge.

Avantages

- Une copie précise de la réalité, qui permet d'obtenir un excellent mouvement.
- La méthode la plus utilisée actuellement pour la production des animations.
- Le réalisme des mouvements obtenus par cette méthode est totalement incomparable avec les autres méthodes.

Inconvénients

- Mais elle a un défaut majeur pour l'édition de mouvement, qui est son inadaptabilité aux physionomies différentes, entre le personnage virtuel, et le personnage réel capturé. Le mouvement capturé est figé dans sa spécificité.
- Les sessions de capture sont très coûteuses et demandent beaucoup de travail et de répétitions.
- N'apporte pas de concept nouveau à la méthodologie de l'animation.

Le tableau 1.1 présente une comparaison entre les différentes technique de génération de mouvement :

Technique	Caractéristique	Avantages	Limites
Par images clés	<ul style="list-style-type: none"> -Dessine les principales images décrivant le mouvement. -Utilise une méthode d'interpolation pour calculer les images intermédiaires. 	<ul style="list-style-type: none"> -La qualité du mouvement résultant est très dépendante du choix des postures clés. 	<ul style="list-style-type: none"> -L'algorithme d'interpolation peut provoquer des effets indésirables. -Le problème se complique lorsque les deux dessins ne sont pas d'une forme proche.
Procédurale	<ul style="list-style-type: none"> - utilise des règles procédurales. - Le mouvement est décrit algorithmiquement. 	<ul style="list-style-type: none"> - Pour créer de nouveaux mouvements, il faut décrire un nouvel algorithme (procédure). 	<ul style="list-style-type: none"> - Les animations sont rarement réalistes.
Simulation physique	<ul style="list-style-type: none"> -la génération des mouvements à l'aide des lois physiques 	<ul style="list-style-type: none"> - Beaucoup réaliste, car les lois sont inspirées de la nature. -Capable de générer des mouvements physiquement corrects. 	<ul style="list-style-type: none"> -Le temps de calcul est très prohibitif. - Nécessite une compréhension approfondie des paramètres physiques.
Capture de mouvement	<ul style="list-style-type: none"> -Traquer et enregistrer les mouvements effectués par l'acteur à l'aide des systèmes informatiques. -Utiliser les marqueurs, les caméras, les récepteurs et les logiciels qui permettent de récupérer les positions de l'acteur. 	<ul style="list-style-type: none"> - La méthode la plus utilisée actuellement pour la production des animations. - Une copie précise de la réalité, qui permet d'obtenir un excellent mouvement. 	<ul style="list-style-type: none"> - Défaut majeur pour l'édition de mouvement, qui est son inadaptabilité aux physionomies différentes, entre le personnage virtuel, et le personnage réel capturé. -Sont très couteuses et demandent beaucoup de travail et de répétitions.

Table 1.1 : Comparaisons des principales Technique de génération de mouvement.

1.6 Conclusion

Dans ce chapitre nous avons essayé de faire une étude sur l'humanoïde virtuel et les modèles, qui lui sont attachés la comparant ces différents modèles. Nous avons constaté que le modèle de couches est le plus proche de la nature humaine.

Après cela nous avons comparé les différentes méthodes de génération de mouvements, nous avons pu conclure que la méthode basé sur la capture de mouvement est la meilleure méthode d'animation de personnage jusqu'à maintenant, car le réalisme d'animation obtenu par cette méthode est totalement incomparable, la plus cette technique est de plus en plus utilisé pour produire rapidement des animations et une copie précise de la réalité, qui permet d'obtenir un excellent mouvement.

III. Chapitre 2 : Capture de mouvements et Edition de mouvements

2.1 Introduction

La prise en compte du mouvement humain est aujourd'hui un enjeu majeur dans le monde de l'informatique. L'augmentation des capacités de calcul accompagne le souhait d'intégrer toujours plus de ressources en animation, ceci afin d'augmenter le degré de réalisme et d'immersion des applications proposées. Qu'il s'agisse d'afficher un joueur de football dans un jeu vidéo, de l'étude de la marche d'un patient dans le cadre d'une étude médicale ou encore de la synthèse d'expressions faciales pour des agents interactifs, il est dans tous les cas nécessaire de disposer de données d'animation de grande qualité, dans des quantités souvent importantes.

Dans ce chapitre, nous allons dans un premier temps abordé les systèmes de capture mouvement et les domaines d'application qui leur sont associés, Parmi tous les systèmes existants nous distinguons trois grandes familles : magnétique, mécanique et optique. Nous allons donc d'une part présenter les systèmes de capture de mouvements avec les avantages et inconvénients de chacune, ainsi que les différentes étapes de capture du mouvement et d'autre part expliquer les fichiers de capture avec leur méthode d'utilisation. Enfin, nous définissons l'édition de mouvement et les différentes techniques qui leur sont liées avec les avantages et les inconvénients de chacune d'entre elles.

2.2 Capture de mouvement

La capture de mouvement (motion capture en anglais, parfois abrégé en mocap) désigne de façon générique les techniques permettant de capter les mouvements d'un élément réel afin de les renvoyer dans un univers virtuel: les mouvements sont enregistrés ou restitués en temps réel vers d'autres systèmes. Le concept général de la capture de mouvement est de filmer un mouvement réalisé par une personne réelle et de le numériser pour le transformer en animation 3D [23]. Cette technique est employée pour faire correspondre des mouvements du vivant à ceux d'un avatar ou d'une représentation conceptuelle.

2.2.1 Domaine d'application

La capture de mouvement a fait de grands progrès ces dernières années. Elle fait l'objet de nombreuses études et applications parmi celles-ci, nous pouvons distinguer :

- **L'industrie du film** : dans ce contexte, l'objectif est d'animer avec réalisme des humanoïdes ou des créatures reproduisant une gestuelle humaine, c'est par exemple le cas de film Avatar ou Le pôle express (figure2.1(a) et (b)).
- **L'industrie des jeux vidéo** : la capture de mouvement humain est utilisée pour obtenir des gestes techniques précis et réalistes difficiles à reproduire par des animateurs, cela concerne par exemple les jeux des sports tels que la série FIFA (figure2.1 (c)). Elle permet également d'acquérir à bon marché un grand nombre d'animations.
- **Domaines d'application connexes** : en dehors de la reproduction du mouvement d'un personnage, l'étude du mouvement humain capturé est également exploitée dans le cadre d'autres types d'applications telles que l'analyse de performance sportives [22], la classification, ou la reconnaissance de gestes (par exemple dans le cadre de la reconnaissance de gestes de langue des signes, la danse ou les gestes musicaux)



Figure 2.1 : Animation de personnages guidés par des données capturées dans les films d'animation et les jeux.

2.2.2 Les systèmes de capture du mouvement

Parmi tous les systèmes existants, on distingue trois grandes familles : magnétique, mécanique et optique. Nous allons présenter dans ce chapitre ces systèmes de capture de mouvements et les avantages et inconvénients qui leur sont liés.

2.2.2.1 La capture mécanique

Les systèmes mécaniques sont des exosquelettes dont chaque articulation est munie d'un potentiomètre qui mesure la position et l'orientation du membre. En général, le squelette est une armature en matériaux composites avec des capteurs linéaires de position et d'orientation reliés entre eux (figure 2.2) [24]. A partir d'un exosquelette constitué de potentiomètres, placés sur le corps d'un être humain, le système capture les angles de flexion entre les articulations des différents membres, on les appelle systèmes prothétiques. Même si ces systèmes semblent imposants, ils sont cependant encore utilisés en raison de leur faible coût, leur fiabilité ainsi que leur facilité d'utilisation [21].

L'exosquelette pour la capture de mouvement du corps n'est pas modulable. En revanche, ce type de système présente l'avantage de pouvoir fonctionner sans fil et dans un espace de travail pouvant atteindre en théorie plus de 300m de rayon autour de la centrale d'acquisition. De plus il ne subit aucune perturbation externe d'ordre magnétique, et peut donc s'utiliser dans toutes les situations.

A chaque acquisition d'une image, la nouvelle position définit une nouvelle « clé d'animation ». La capture de la position se fait en temps réel, image par image. L'orientation des membres est calculée en temps réel. Un récepteur de position et d'orientation indique le déplacement et l'orientation du corps dans l'espace. La fréquence d'échantillonnage varie suivant les systèmes de 30Hz à 240Hz.

Les avantages de ce type de système sont les suivants :

- Prix par rapport aux autres systèmes et la facilité d'utilisation puisqu'ils n'exigent pas des opérateurs spécialisés à la différence des systèmes magnétiques ou optiques [24]. Presque aucun post-traitement et très peu de puissance de calcul sont nécessaires, ce qui rend leur utilisation dans un environnement en temps réel possible.
- une phase de traitement simplifiée (aucune occultation et bruit minime sur les mesures) [25].

Les inconvénients de ce type de système sont les suivants :

- L'encombrement du système sur le corps de l'acteur, tel que la contrainte produite à cause du squelette qui doit être adapté sur le corps (figure2.2) et qui limite les mouvements de l'acteur. Ils ne sont pas recommandés pour la capture de mouvements rapides et/ou volumineux [21] [24],
- Ils ont une faible fréquence d'acquisition et posent problème pour le calcul de la translation globale.



Figure 2.2 : System mécanique GypsyGyro (société Gypsy)

2.2.2.2 La capture magnétique

Le principe de fonctionnement des systèmes magnétiques est maintenant donné :

Les récepteurs munis du même système d'antennes déterminent leur position et orientation par rapport à l'émetteur par l'intermédiaire d'une unité de traitement qui leur est propre.

Placés sur un objet mobile les récepteurs renvoient les informations de transformations du mobile par rapport à l'émetteur.

L'émetteur et les récepteurs sont reliés à un boîtier ou une carte qui communique avec l'ordinateur via le port parallèle, le port série ou Ethernet (figure 2.3). Les données sont envoyées à l'ordinateur [21] [24].

Les avantages de ce type de système sont les suivants

- Position et orientation des capteurs disponibles sans traitement.
- Capture presque temps réel du personnage virtuel animé, car les données des capteurs sont traitées en temps réel.
- Aucune occultation de captures

Les inconvénients de ce type de système sont les suivants :

- Difficile de changer le positionnement des captures.
- Contraintes sur l'environnement (faible zone d'acquisition).
- Contraintes sur sujets (câble, capteurs).
- Fréquence d'acquisition insuffisante pour certains mouvements

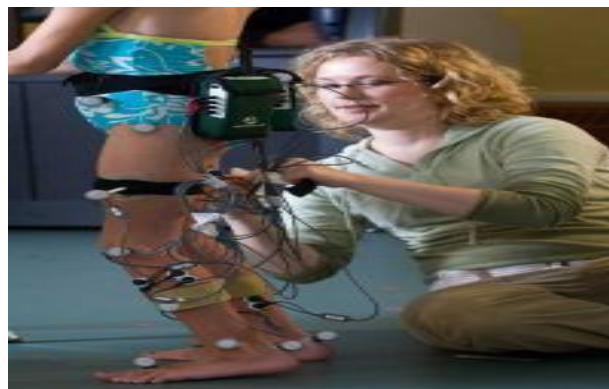


Figure2.3 : exemple de la capture magnétique.

2.2.2.3 Système optique

C'est le type de système le plus courant, utilisé depuis des années dans la création de mouvement de personnages virtuels, il est basé sur une technologie optique qui emploie des données captées par les caméras et la position des objets dans l'espace 3D. Les caméras suivent des points prédéterminés (marqueurs) sur les parties du corps de l'acteur. Les marqueurs peuvent être soit passifs (réfléchissant) ou actifs (émettant de la lumière) [26].

Les caméras sont disposées en cercle autour du volume de travail. Le nombre de caméras influe sur la précision et le temps de traitement. Généralement, on utilise 6 à 32 caméras pour les mouvements du corps et 1 à 2 caméras pour l'animation faciale. Les marqueurs sont de petites sphères réfléchissantes. La position des sphères dans l'espace est calculée à partir des images produites par les caméras. Le centre de gravité de chaque sphère détermine la position du mobile. L'orientation est calculée à partir de 3 marqueurs formant un triangle. La taille des marqueurs varie suivant le volume de travail et le type de capture [24].

Il existe deux grandes catégories dans cette technologie, que nous détaillons dans ce qui suit, les systèmes optiques sans marqueur et avec marqueur :

a) Systèmes optiques sans marqueur

L'estimation du mouvement humain à partir de systèmes optiques sans marqueur est un problème complexe. Inférer la pose d'un humain uniquement à partir d'images est équivalent à un problème de recherche dans l'espace des paramètres d'un modèle d'humanoïde choisi. Il existe certains facteurs qui font que ce problème soit difficile :

- Premièrement le corps humain est une structure complexe et très articulée. Même les modèles cinématiques simples du corps humain contiennent plus de trente degrés de liberté. Ce fait rend la recherche de solution difficile, surtout lorsque l'on souhaite acquérir les mouvements fins du corps humain.
- Un second facteur impacte sur les difficultés algorithmiques des approches optiques sans marqueur.

Cette approche possède l'avantage concernant le faible coût de matériel et un environnement peu contraint, mais elle est limitée dans le cas d'une analyse complexe de l'image et lors des Occultations.

b) Systèmes optiques à base de marqueurs

Les systèmes commerciaux d'acquisition de mouvements les plus utilisés sont des systèmes optiques à base de marqueurs. Ces approches utilisent le principe des points éclairés en mouvement, initialement proposé par Johansson [27]. Des marqueurs optiques, composés d'un matériau hautement réflecteur ou encore de Diodes Electro-Luminescentes, sont placés sur le sujet à suivre. Plusieurs caméras spécifiques à haute vitesse (parfois associées à une source de lumière caractéristique) sont utilisées pour filmer les personnes en mouvement. Les approches optiques à base de marqueurs ne sont pas seulement utilisées pour la capture du mouvement humain, mais aussi pour des animaux, des objets, etc.

b.1) Marqueurs actifs

Ces systèmes à base de Marqueurs actifs (ex. Selspot, Watsmart, Optotrak) utilisent des caméras infrarouges pour enregistrer la position de diodes émettrices de lumière (LED) placées sur des points anatomiques particuliers. Chaque marqueur s'identifie par les paramètres des ondes émises (longueur d'onde et fréquence). Ils se différencient les uns des autres. Ces marqueurs sont donc identifiés automatiquement et immédiatement en temps réel comme étant des points dynamiques avec des coordonnées 3D dans l'espace virtuel (figure 2.4).

L'avantage principal est la liberté de mouvement est très peu altérée par des marqueurs petits et légers. Cependant, un des inconvénients majeurs est le non différenciation des marqueurs.

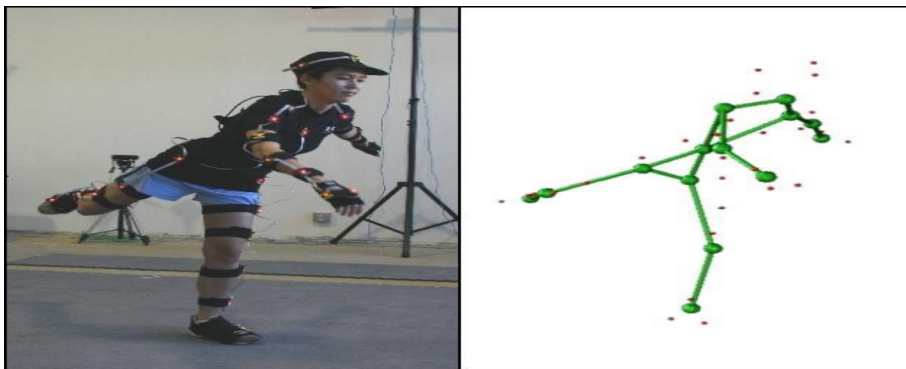


Figure 2.4 : exemple de marqueurs actifs.

b.2) Marqueurs passifs

La technologie de capture de mouvement de Vicon à base de Marqueurs passifs (ex. Vicon, Elite, Motion Analysis) utilise des marqueurs recouverts de Scotch-Lite (figure 2.5).

Ils ont la propriété de renvoyer la lumière infrarouge des caméras sous l'angle incident. La fréquence d'acquisition varie le plus souvent entre 50 et 240 Hz. Le principe de fonctionnement est similaire du radar : la caméra lance des radiations (généralement infrarouge) qui sont reflétées par les marqueurs et retournent à la même caméra. Les caméras sont sensibles à une bande limitée de longueurs d'onde et perçoivent les marqueurs comme des spots lumineux. Ces systèmes se composent généralement de 6 à 24 caméras. Les systèmes passifs n'exigent pas de l'utilisateur de porter des fils ou des équipements électroniques. Ils sont généralement attachés directement à la peau.

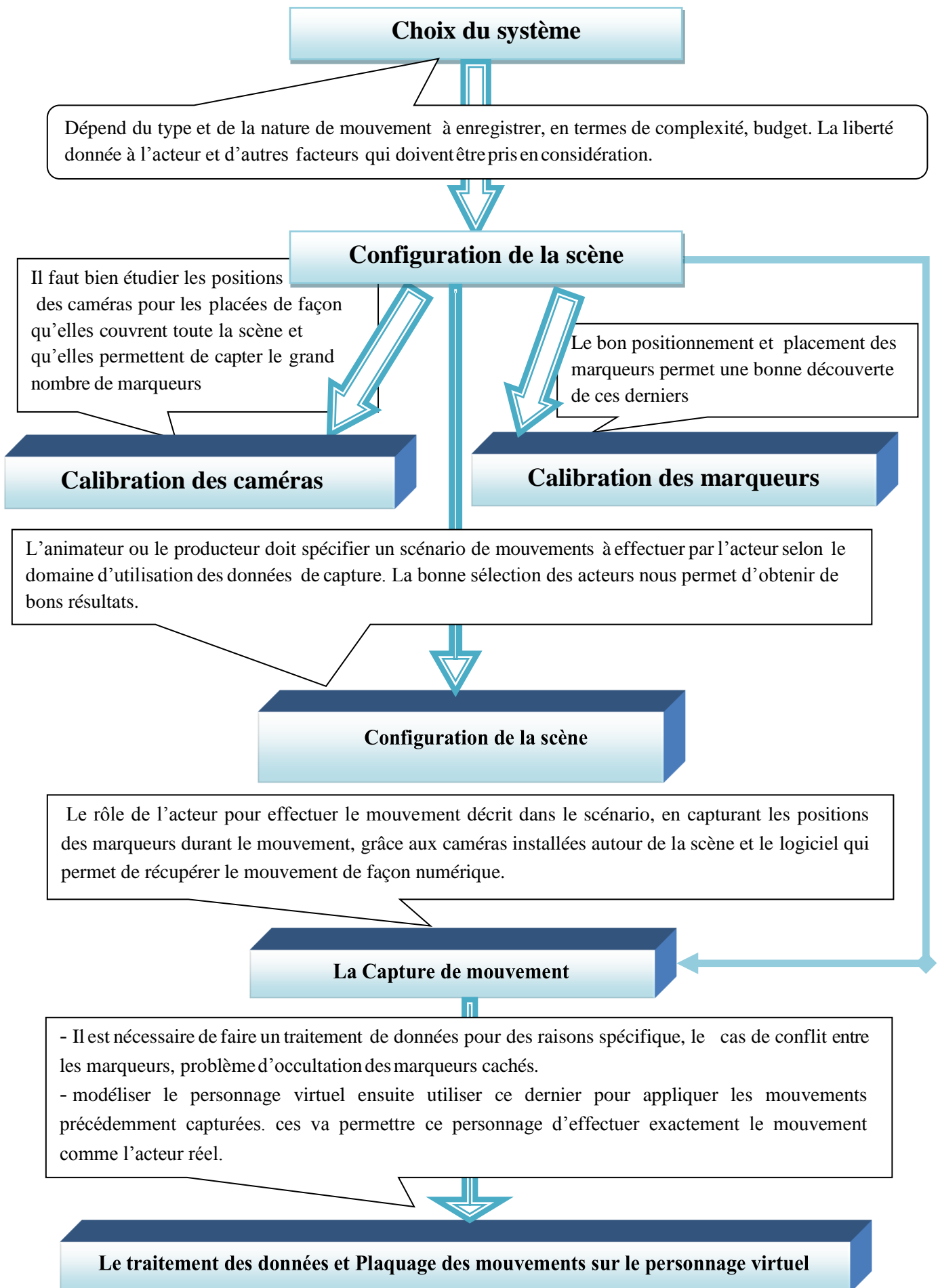


Figure 2.5 : Exemple de marqueurs passifs.

Cette technologie possède de nombreux avantages et nous citons les plus importants tels que la facilité de changement de la configuration des marqueurs, la possibilité d'estimation du squelette interne grâce à des groupes de marqueurs, la très grande fréquence d'acquisition possible, le champ d'acquisition relativement grand (dépendant du nombre de caméras), un sujet faiblement contraint dans ses mouvements par le système, la grande précision des données acquises dans la plupart des cas et le grand nombre de marqueurs pouvant être utilisés. Parmi leurs limites importantes figurent les traitements coûteux liés aux nommages et occultations des marqueurs, l'impossibilité de leur utilisation dans le cas de systèmes à base de lumière infrarouge et un long temps de retouche (traitement par software) [24].

2.2.3 Processus de capture du mouvement

Il existe plusieurs étapes pour obtenir un fichier de capture de mouvement de bonne qualité, les différentes étapes de la capture du mouvement sont résumées par schéma comme suit :



2.2.4 Fichiers des mouvements capturés

Les formats de fichiers de mouvement et geste sont aujourd'hui largement utilisés dans de nombreuses applications qui traitent avec le mouvement et le signal de geste. C'est le cas dans des domaines comme la capture de mouvement, l'animation de personnages, l'analyse de geste, de la biomécanique, les interfaces gestuelles musicales, la chirurgie virtuelle. Ces formats sont soit des formats de bas niveau, soit près de formats le signal produit par le système de capture.

Il existe différentes sociétés qui ont développé de nombreuses extensions de fichiers contenant les mouvements capturés, on note les extensions les plus utilisés BVA et BVH produites par la même société Biovision, et C3D par Biomechanics, HTR par Motion Analysis.

2.2.4.1 Fichiers BVH (Biovision Hierarchical Data)

C'est un format de fichier d'animation de personnages développé par Biovision, une société spécialisée en capture de mouvement, pour donner les données de capture de mouvement à leurs clients. BVH est largement utilisé.

a) Structure du fichier BVH

Le fichier BVH est composé de deux parties, la première nommée HIERARCHY qui représente la construction du squelette et la position des articulations, lorsque le squelette est en pose initiale. Une deuxième partie nommée MOTION qui contient des transformations au cours du temps pour chaque frame [28].

La représentation du squelette hiérarchique est illustrée dans la figure suivante. Une illustration de la position initiale (frame 0) et le frame suivant de l'animation est représentée par la figure suivante.

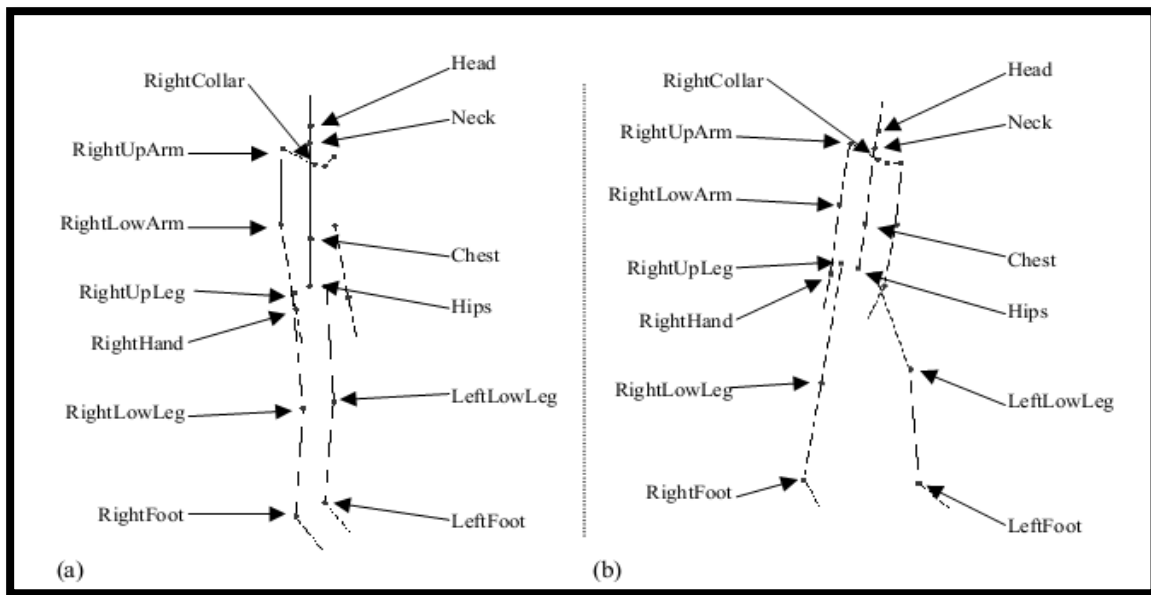


Figure 2.6 La structure du squelette d'un fichier d'exemple BVH, (a) la position initiale, (b) frame suivant.

b) La partie hiérarchique :

La section hiérarchique du fichier commence par le mot-clé HIERARCHY, elle contient une description détaillée du format du squelette, Dans la ligne suivante, on trouve le mot ROOT suivi par le nom de la racine du squelette qui est généralement le bassin (Hips), Le mot clé OFFSET définit la translation correspondante par rapport à l'articulation père ou globalement dans le cas de la racine le long des axes X, Y et Z respectivement. Le reste du squelette est défini de manière récursive, et chaque fils est encapsulé par '{' et délimité dans la ligne précédente par le mot clé JOINT suivi par le nom de la jointure fils.

La deuxième ligne de chaque définition d'une jointure contient le mot clé CHANNELS qui définit le degré de liberté de la jointure courante représenté par la première valeur qui suit ce mot, elle peut varier de 1 à 6 au maximum. La même ligne, contient les différents types de transformations appliquées sur l'articulation à chaque frame, dont Xposition, Yposition et Zposition correspond aux translations, et Xrotation, Yrotation et Zrotation correspond aux rotations sur les axes X, Y et Z. On note qu'on peut trouver des fichiers qui contiennent plus qu'un seul squelette dans les cas de multiples acteurs pour cela on trouve la redondance du mot ROOT. Il faut respecter l'ordre spécifié dans la section CHANNELS car il est identique aux données dans la deuxième partie du fichier, et à l'ordre de

concaténation des angles d'Euler, lors de la création de la matrice de rotation de chaque jointure par l'équation suivante.

$$R = R_z R_x R_y$$

Le caractère '}' termine la déclaration de la jointure courante. Pour signifier la fin d'une branche, le mot clé End Site et décrit qui encapsule l'OFFSET.

```

HIERARCHY
ROOT Hips
{
  OFFSET 0.00000 0.00000 0.00000 /* L'offset de la jointure par rapport au père*/
  CHANNELS 6 Xposition Yposition Zposition Zrotation Yrotation Xrotation
  JOINT LHipJoint
  {
    OFFSET 0 0 0
    CHANNELS 3 Zrotation Yrotation Xrotation
    JOINT LeftUpLeg
    {
      OFFSET 1.65674 -1.80282 0.62477
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT LeftLeg
      {
        OFFSET 2.59720 -7.13576 0.00000 /* Les degrés de libertés jointure*/
        CHANNELS 3 Zrotation Yrotation Xrotation
        JOINT LeftFoot
        {
          OFFSET 2.49236 -6.84770 0.00000
          CHANNELS 3 Zrotation Yrotation Xrotation
          JOINT LeftToeBase
          {
            OFFSET 0.19704 -0.54136 2.14581
            CHANNELS 3 Zrotation Yrotation Xrotation
            End Site
            {
              OFFSET 0.00000 -0.00000 1.11249
            }
          }
        }
      }
    }
  }
}

```

Déclaration d'une jointure LeftUpLeg(Jambe Gauche).

La fin du branche avec le mot clé End Site.

Figure 2.7 Exemple de la partie hiérarchique du fichier BVH.

c) La partie mouvement

La deuxième section d'un fichier BVH, qui est désigné par le mot-clé MOTION, dont la ligne suivante contient le mot clé Frames : suivi par le nombre de frames composants

l'animation. La vitesse de défilement est sur une ligne commençant par "Time Frame:" qui est suivie par un flottant positif que représente la durée d'une chaque frame en secondes.

Dans le reste de cette partie on trouve une liste de nombres flottants correspondant aux transformations à faire dans l'ordre spécifié dans la partie de hiérarchique [28]. Chaque ligne représente un frame, comme indiqué dans la figure suivante.

MOTION								
Frames:	2							
Frame Time:	0.033333							
8.03	35.01	88.36	-3.41	14.78	-164.35	13.09	40.30	} ligne représente un frame
35.10	86.47	-3.78	12.94	-166.97	12.64	42.57	-22.34	

Figure 2.8 Exemple de la partie mouvement du fichier BVH

2.3 Edition de mouvement

Comme un personnage doit pouvoir bouger, il faut bien sûr créer son mouvement. Plusieurs possibilités s'offrent à nous : soit en utilisant un mouvement réel, et mesurer son évolution, pour en faire une copie, soit en analysant le mouvement, pour extraire ses composantes et ensuite les combiner pour en produire de nouveaux, soit en « dessinant » le mouvement. L'animateur ne peut pas utiliser immédiatement les mouvements capturés, il a besoin des techniques pour éditer le mouvement.

L'édition de mouvement est une technique qui permet d'obtenir un nouveau mouvement, en se basant sur un mouvement qui existe déjà (base de mouvement), afin de corriger, modifier ou adapter le mouvement capturé. Lors de la modification de mouvement on doit préserver quelques caractéristiques du mouvement original afin d'obtenir un mouvement réaliste et contenu. Il existe plusieurs méthodes d'édition de mouvement,

2.3.1 Méthodes d'édition de mouvement

Il existe plusieurs méthodes d'édition de mouvement tel que le mélange des mouvements «Motion Blending », Déformation cinématique « Motion Warping », adaptation de mouvements « Motion Retargeting », spatio-temporelle «Spacetime Constraints », cinématique inverse «Inverse kinematics » et d'autres méthodes comme un mélange de deux ou de plusieurs méthodes précédentes [42].

a) Mélange de mouvements « Motion Blending »

Les méthodes de mélange des mouvements cherchent à obtenir un mouvement par composition de deux ou plus de mouvements capturés. Par exemple un mouvement de marche et de saut, vu qu'il est impossible de capturer les deux mouvements en même temps mais séparément, on peut vouloir les enchaîner (**figure 2.9**) Pour cela il faut faire une transition entre la fin du mouvement de marche et le début du mouvement de saut. On utilise les techniques de mélange de mouvements pour réaliser cette transition.

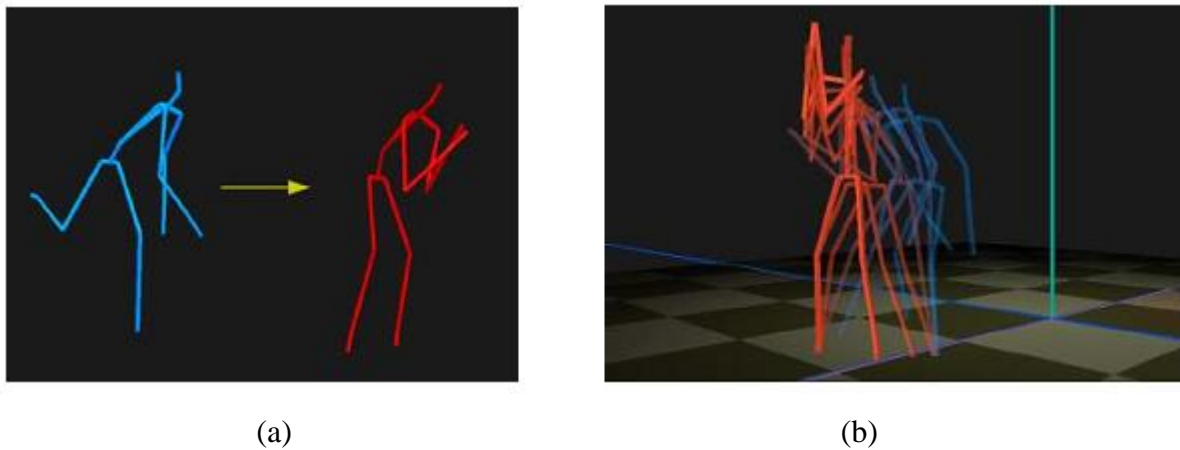


Figure 2.9 (a) les deux positions à mélanger (b) l'animation de transition entre les ces deux positions.

Les premiers travaux dans ce domaine proposèrent des interpolations linéaires dans une base de données de trajectoires capturées [29]. Une autre approche consiste à raisonner posture par posture plutôt que de considérer un mouvement dans sa globalité. Ainsi, des graphes de mouvements sont proposés et ont été introduits pour représenter une transition entre deux mouvements compatibles. Deux postures de deux différents mouvements peuvent être connectées dans ce graphe si elles sont compatibles [30] : chaque nœud correspond à une posture et chaque arc est une transition possible entre 2 postures .

Les problèmes lors du mélange des mouvements : Les durées des mouvements sont différentes. Les temps choisis pour certains événements sont différents. Supposons que nous mélangeons deux mouvements de marche où les temps sont différents. On a besoin de synchroniser les deux mouvements (**figure 2.10**)

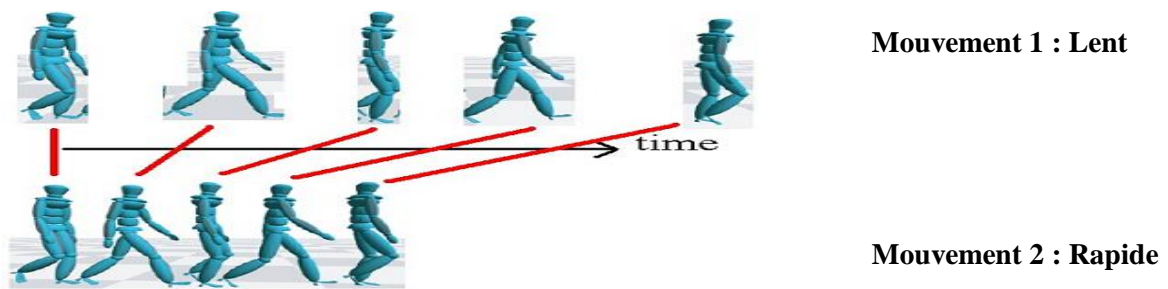


Figure 2.10 Deux mouvements des marches avec temps différentes.

b) Spatio-temporelle «Space time Constraints »

La méthode de «Space time constraint » peut être utilisée pour synthétiser des nouveaux mouvements ou pour éditer des mouvements déjà existants, en utilisant des contraintes dynamiques ou physiques. En plus, la formulation du spatio-temporel est l'origine de beaucoup de travaux dans l'animation des humains virtuels.

Les premiers travaux dans ce domaine proposèrent de déformer un mouvement de référence en utilisant une technique d'optimisation visant à trouver le meilleur mouvement satisfaisant un ensemble défini de contraintes [31]. L'optimisation des contraintes spatio-temporelles offre une méthode générale pour déformer un mouvement d'origine. Mais il existe beaucoup de facteurs qui limitent l'utilisation de cette technique.

Il y a de nombreuses équations de contraintes et de variables indépendantes du fait de l'optimisation du système sur un intervalle de temps (et non plus sur une seule posture comme la cinématique inverse).

La complexité du système (coût de calcul) impose une limitation sur l'échantillonnage de l'intervalle (nombre de postures prises en compte) ainsi que sur la complexité articulaire du sujet animé (nombre de degrés de liberté du squelette).

L'utilisateur doit spécifier manuellement l'intervalle au cours duquel le mouvement est susceptible d'être déformé. Si celui-ci est mal choisi, une solution incorrecte peut être trouvée. Un trop grand intervalle peut engendrer un ensemble de solutions trop important pouvant induire une convergence du système dans un minimum local non désiré. A l'inverse, un intervalle trop petit peut restreindre de manière trop importante l'espace de solutions, rendant impossible la résolution des contraintes.

La prise en compte du mouvement sur un intervalle de temps empêche l'utilisation de cette technique lors d'animations en temps-réel.

Pour minimiser le coût de calcul, [32] ont défini des fenêtres spatio-temporelles permettant de diviser le problème (et donc la complexité) en un ensemble de sous-problèmes et de fournir aux animateurs une rétroaction interactive. Plusieurs propositions dans ce domaine pour minimiser le coût et augmenter la fiabilité de cette méthode comme l'application des lois de Newton à un humanoïde simplifié pour minimiser le coût de calcul.

c) Adaptation de mouvements « Motion Retargeting »

Les techniques d'adaptation de mouvement (Retargeting) consistent à adapter un mouvement connu d'un personnage à un autre. Cette technique est utilisée pour ajouter de petits changements à un mouvement.

Une vision simple de l'adaptation de mouvement [33]. Consiste à reproduire le mouvement d'un personnage à un autre de proportions différentes tout en conservant certaines contraintes importantes.

Le principe de l'adaptation d'un mouvement est de modifier un mouvement capturé afin de résoudre des contraintes cinématiques ou dynamiques différentes, comme par exemple modifier un mouvement de préhension pour saisir un objet dont la position est différente par rapport à celle de la trajectoire capturée. Pour résoudre ce type de problème, une approche consiste à imposer des contraintes (position à un instant donné) et à utiliser des méthodes d'optimisation non-linéaire pour trouver une solution [31] [34]. Ces méthodes se limitent généralement à des contraintes cinématiques, même si quelques travaux tentent de tenir compte partiellement de la dynamique [35].

Ainsi, Ménardais et Multon [36] proposent une architecture logicielle permettant à la fois de combiner et d'adapter des mouvements capturés. À partir de cette architecture, « Kulpa » s'intéresse à l'adaptation de mouvements pour des contraintes cinématiques, dynamiques et cinétiques pour proposer une méthode de simulation de mouvements appelée MKM (Manageable Kinematic Motion) [37].

Enfin, il peut également être nécessaire d'adapter les trajectoires articulaires (obtenues par capture de mouvement) pour un nouveau squelette de morphologie différente. Plusieurs

travaux s'intéressent à cette phase d'adaptation : Monzani et al [38] utilisent un squelette intermédiaire pour adapter des trajectoires articulaires à une nouvelle morphologie, Choi et Ko proposent une adaptation temps réel des mouvements d'un personnage à un autre [39] ou encore Ménardais et al proposent une adaptation de trajectoires capturées pour un nouveau mannequin de morphologie différente ou pour un nouvel environnement.

Enfin, Pronost et al [40]. proposent une interpolation, à partir d'une base de données de mouvements capturés, en fonction des paramètres anatomiques d'une personne à simuler [40]. Yamane et al. proposent une approche d'animation de personnage manipulant des objets combinant un planificateur de trajectoires et une base de données de mouvement capturés [41].

d) Déformation cinématique « Motion warping »

La première solution pour déformer un mouvement consiste à modifier les orientations des articulations du squelette permettant d'obtenir une nouvelle posture. Witkin et Popovic [48] ont ainsi décrit une méthode appelée « motion warping » pour éditer et modifier un mouvement cinématique de référence afin d'obtenir un nouveau mouvement adapté. Cette adaptation est effectuée en spécifiant une série de contraintes cinématiques sur les trajectoires des paramètres appelées courbes de mouvement «motion curves». Dans un premier temps, toutes les postures clés sont déterminées de manière à vérifier les contraintes et à se rapprocher au mieux de la posture d'origine voir (figure 2.11). Le nouveau mouvement est alors constitué de ces postures-clés. Cette méthode est simple à réaliser et bonne pour éviter les obstacles, mais crée des problèmes sur les segments de corps humain virtuel.

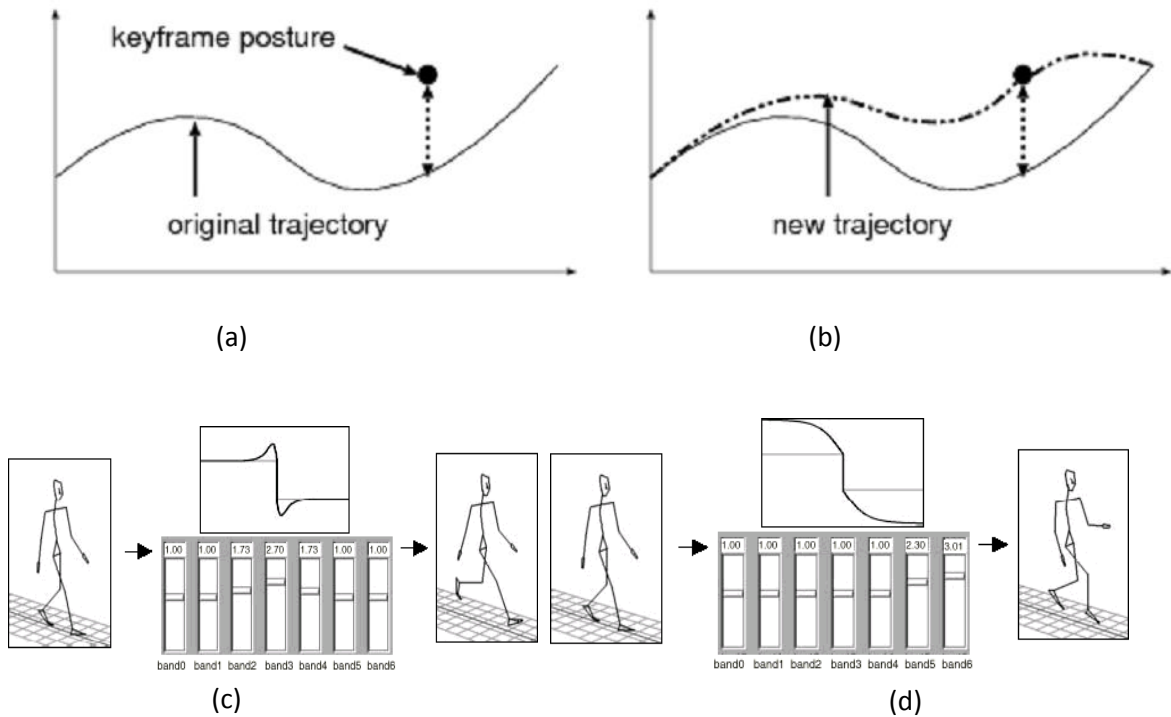


Figure 2.11 les étapes Déformation cinématique (a) mouvement original + postures-clés (b) nouveau mouvement. (c) modification des moyennes fréquences (d) modification des basses fréquences.

e) Cinématique inverse «Inverse kinematics»

La cinématique inverse (souvent abrégée IK, de l'anglais inverse kinematics) est un procédé par lequel on peut déterminer les positions et rotations d'articulations d'un modèle afin d'obtenir une pose [60]. Cinématique inverse donc est un procédé d'animation et d'articulation qui permet de trouver les valeurs des angles d'une chaîne cinématique afin de placer cette chaîne dans une certaine posture, répondant à un certain nombre de contraintes (figure 2.12).

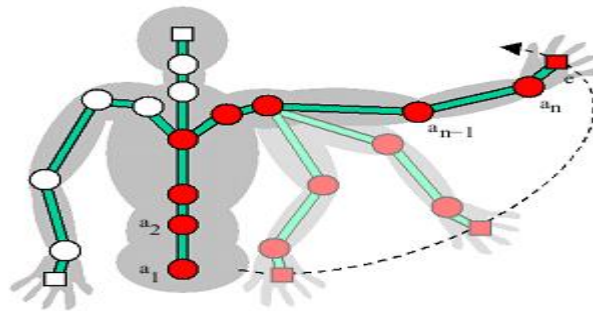


Figure 2.12 : Exemples de contrôle de chaîne articulée par cinématique inverse

Cinématique inverse et applications les techniques de cinématique inverse il a étudié à grande échelle au cours des dernières décennies, dans le domaine graphique [43] mais surtout robotique [44]. Ces méthodes restent un outil de choix, à la fois pour édité des mouvements a priori inconnus, mais aussi comme composant de méthodes plus spécifiques comme l'adaptation de mouvements à de nouvelles contraintes [46], ou bien encore la reconstruction de mouvements partiellement acquis depuis des méthodes de capture [45]. Les approches les plus célèbres ont fait l'objet d'études comparatives dans le cadre de la reconstruction de différentes classes de mouvement [47].

Cependant, le mauvais conditionnement mathématique du problème, couplé à l'importante redondance des chaînes articulées considérées (que ce soit pour un bras ou une jambe, mais à plus forte raison pour un humanoïde complet), nécessite l'adjonction de tâches secondaires qui viennent sur contraindre le problème [49]. Nous allons parler en détail de cette méthode en spécifiant son principe et ses types dans le chapitre suivant.

2.4 Conclusion

La capture de mouvements présente un intérêt clair et le réalisme des animations obtenues est incomparable avec les méthodes d'animation classiques comme les méthodes cinématiques ou dynamiques. Cependant cette technique présente un défaut qui est la contre partie directe de sa force : le mouvement capturé est figé dans sa spécificité, et si on tente de le modifier pour l'adapter à une morphologie différente, ou le déformer pour répondre à un objectif donné, le réalisme du mouvement est altéré de manière significative [50].

Différentes méthodes d'édition de mouvement (présentées dans ce chapitre) répondent partiellement à cette lacune et tentent de se servir du mouvement capturé comme d'une base modifiable pour générer des animations particulières.

IV. Chapitre 3: Technique d'édition de mouvement basée sur la cinématique inverse

3.1 Introduction

Comme nous l'avons vu au **chapitre 1**, 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. Pour générer une nouvelle pose avec la position de but connue et contrôler les structures articulées de manière complexe, la cinématique inverse (IK) est l'une des techniques les plus utilisées.

Dans ce chapitre, nous allons donner un aperçu des méthodes les plus importantes à la résolution de problème de la cinématique inverse. Les méthodes les plus connues et les plus utilisées sont les méthodes analytiques, méthodes itératives, méthodes basés sur le gradient, méthodes de Descente de Coordonnées Cycliques (Cyclic Coordinate Descent) (CCD) et FABRIK (Forward and Backward Reaching Inverse Kinematic)

3.2 Cinématique directe

En animation par ordinateur, la cinématique directe consiste à donner une pose à un squelette d'animation, en modifiant uniquement les paramètres de ses articulations (variables d'articulation). Pour une série d'articulations inter-reliées, chaque articulation est affectée individuellement. Pour obtenir une pose précise, il faut donc manuellement définir la rotation (angle) de chacune des articulations, l'interdépendance entre les articulations (parent et enfant) peut rendre le processus laborieux.

Il suffit donc de décrire la manière dont doivent évoluer les différents angles. Il existe de nombreuses manières efficaces et numériquement correctes pour définir ces angles. Sous forme d'équation, cela revient à résoudre :

$$\mathbf{X} = \mathbf{f}(\boldsymbol{\theta}) \quad (3.1) \quad \text{Avec :}$$

$\mathbf{X} = (X_x, X_y, X_z, X_r, X_q, X_y)$: position terminale d'un élément du squelette

$\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \dots)$: vecteur des angles d'articulation.

Cette méthode est simple à mettre en œuvre. Cette simplicité du traitement nécessaire à son application permet de manipuler les humanoïdes en temps réel, offrant ainsi une prévisualisation du résultat final très intuitive. Malheureusement sa manipulation devient très complexe lorsque les chaînes hiérarchiques du squelette deviennent trop longues. En effet, la modification d'un angle d'un des jointures modifie la position de tous les éléments inférieurs dans la hiérarchie. Pour éviter ce problème, une autre technique est disponible : la cinématique inverse.

3.3 Cinématique inverse

La cinématique inverse s'oppose à la cinématique directe. La cinématique inverse joue un rôle clé dans l'animation par ordinateur et la simulation des corps articulés. Souvent, ces corps articulés contiennent plus d'une centaine de degrés de liberté.

La cinématique inverse qui permet de trouver les valeurs des angles d'une chaîne cinématique afin de placer cette chaîne dans une certaine posture, répondant à un certain nombre de contraintes. Mathématiquement, le problème du géométrique inverse est régi par l'équation suivante :

$$\mathbf{q} = \mathbf{f}^{-1}(\mathbf{x}) \quad (3.2)$$

Où $f^{-1}(X)$ représente un système de n équations non-linéaires avec n inconnus.

Dans un problème de cinématique inverse, on cherche à déplacer un effecteur (typiquement, l'extrémité de l'articulation) vers un certain point x . La chaîne du squelette d'animation peut-être représentée par un vecteur \mathbf{q} décrivant les positions (axiales, angulaires) des différentes parties intervenant dans le mouvement (figure 3.1).

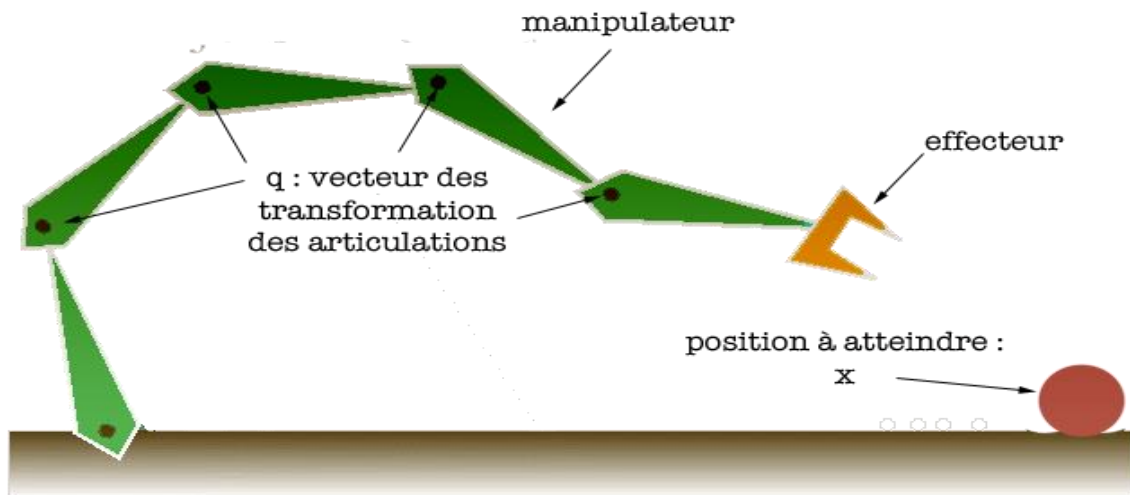


Figure 3.1 : Principe de la cinématique inverse.

3.4 Méthodes de résolution de la cinématique inverse

3.4.1 Méthodes analytiques

Pour gérer une chaîne cinématique simple qui contient quelques degrés de liberté, les méthodes d'analyse sont le meilleur choix [51]. Ces méthodes sont rapides, mais elles ne donnent pas la solution au complexe articulé (comme l'humain) et nombre d'inconnus, aussi à ne pas prendre en compte certaines contraintes. Ce sont des méthodes qui donnent des résultats algébriques minimaux. Elles se basent sur le calcul mathématique direct pour la résolution des équations qui modélisent le système.

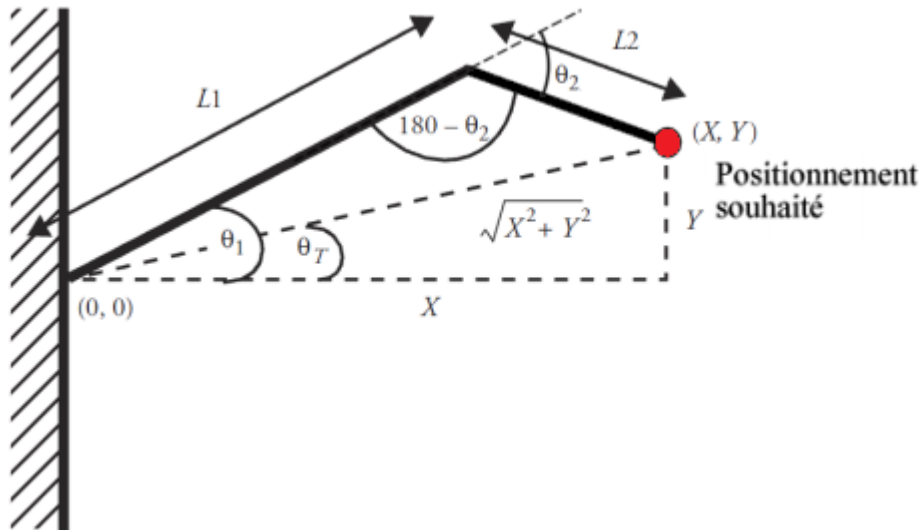


Figure 3.2: Les angles des articulations permettant d'obtenir une telle configuration.

3.4.2 Méthodes itératives

Ce sont des méthodes algorithmiques utilisées pour résoudre un problème d'optimisation. Elles sont considérées comme des procédés itératifs, puisqu'en commençant par une valeur initiale. On converge vers la solution optimale après une succession d'itérations et chaque itération nous donne une solution approximative. 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 [52].

a.1) Méthodes basées sur la Jacobienne

Dans les faits, la plupart des systèmes de cinématique inverse sont trop complexes pour être résolus avec la méthode analytique. On peut cependant déterminer la solution au système de façon itérative en utilisant un Jacobien.

Les méthodes les plus courantes en termes de cinématique inverse font intervenir la matrice Jacobienne J du système (équation 3.3).

La matrice Jacobienne représente les dérivées premières du système pour chaque degré de liberté, c'est-à-dire la direction dans laquelle vont évoluer les effecteurs, si on applique une légère variation aux valeurs des degrés de liberté. La matrice Jacobienne inverse (équation 3.4)

permet une réciproque qui nous permet de déterminer dans quelle direction faire varier les angles de degrés de liberté pour que l'effecteur se déplace dans la direction voulue (la cible).

$$\Delta X = J \Delta q \quad (3.3)$$

Avec : **J** : matrice jacobienne carré régulière du système.

X : situation de l'effecteur terminal.

q : configuration articulaire du squelette.

Pour trouver Δq , on peut exprimer l'équation de cette manière :

$$\Delta X = J^{-1} \Delta q \quad (3.4)$$

Le problème est que la matrice **J** est souvent non inversible, à cause des dimensions différentes de **X** et de **q**. En effet pour une configuration des effecteurs finaux. Malheureusement le calcul de l'inverse d'une matrice quelconque est un problème n'ayant pas de solution dans beaucoup de cas concernant la cinématique inverse. On utilise par conséquent d'autres méthodes donnant des solutions approchées mais acceptables. Ces solutions sont nombreuses, on peut citer :

a.2) La Jacobienne transposée

L'utilisation de la matrice transposée c'est une autre solution du problème, au lieu d'inverser une matrice, on se contente de la transposer cela donne des résultats rapides mais de moins réaliste. La Jacobienne transposée est une méthode de résolution généralement utilisée dans le cas de problèmes ne disposant que d'une seule extrémité.

a.3) La pseudo-inverse

La notion de pseudo-inverse est utilisée en mathématique pour généraliser la notion de matrice inverse dans des cas non inversibles. Les méthodes fondées sur le calcul de la pseudo-inverse J^+ permettent d'inverser l'équation, Cela donne l'équation suivante :

$$\Delta \theta = J^+ \Delta X + \alpha (I + J^+ J) \Delta z \quad (3.5) \text{ avec : } \begin{array}{l} J^+ \text{ pseudo inverse de la matrice jacobienne } J \\ I \text{ matrice identité} \\ \alpha \text{ constante de pénalité} \\ \Delta z \text{ contrainte à minimiser qui correspond à la} \\ \text{tâche secondaire (par exemple contrainte prenant} \\ \text{en compte les angles limitent pour minimiser un} \\ \text{critère énergétique).} \end{array}$$

Nous pouvons donc évaluer les différents angles respectant la position à atteindre de l'effecteur terminal, tout en essayant de minimiser la violation de la contrainte ainsi introduite. Le problème est que cette méthode est assez lourde à calculer, et peut être instable devant certaines singularités.

3.4.3 Méthode basée sur le gradient

a.1) Descente de gradient

Cette méthode [53] s'appuie sur la dérivée de la fonction objectif, ou au moins sur une approximation de celle-ci et éventuellement des dérivées successives comme la méthode de Newton. Elle consiste à chaque itération, à trouver un point voisin, meilleur que la solution courante. Pour trouver ce voisin on se déplace d'un certain pas, avec une certaine direction calculée à partir de la dérivée. On trouve l'optimum global, dans le cas des fonctions objectifs convexes. Sinon elle ne trouve que l'optimum local puisqu'elle ne peut s'échapper d'un optimum local. Seuls certains critères imposés sur le pas de déplacement, et sur le choix de la direction à emprunter, permettent de trouver l'optimum global pour des fonctions qui ne sont pas trop irrégulières.

C'est une méthode d'optimisation considérée comme étant la plus facile et la plus simple. Elle permet de converger vers la solution qui est le minimum local après un certain nombre d'itérations et en commençant par une valeur aléatoire. Tout cela se fait en suivant l'algorithme suivant :

1. Choisir une valeur initiale x_0 et un seuil $\varepsilon \geq 0$.
2. Définir une suite $x_1, x_2, \dots, x_k, x_{k+1}$ par itération en suivant les étapes suivantes :
 - a) Calcul du gradient noté $\nabla f(x_k)$
 - b) Test d'arrêt : si $\|\nabla f(x_k)\| \leq \varepsilon$, arrêt. sinon
 - c) Calcul de $\alpha_k > 0$
 - d) Nouvel itéré $x_{k+1} = x_k - \alpha_k \cdot \nabla f(x_k)$

a.2) Méthode Monte-Carlo

C'est une méthode d'approximation connue par l'utilisation du hasard pour résoudre les problèmes de calculs. Elle est basée sur l'idée de prélever des solutions aléatoires sous forme d'échantillons. Cet échantillonnage va être déplacé suivant le mouvement (qui est à son tour aléatoire) afin de trouver l'optimum. Alors plus le tirage aléatoire des solutions est grand, l'algorithme a plus de chances de converger vers la bonne solution et c'est ce qui peut alourdir le calcul [54].

3.4.4 Méthode Descente de Coordonnées Cycliques CCD

À partir des nombreuses techniques d'optimisation qui existent pour résoudre ce problème de cinématique inverse, Luenberger a proposé une méthode appelée CCD (Cyclic Coordinate Descent). Elle a été utilisée par Wang et Coll. [55] dans le domaine de la robotique et une version proche du CCD a par ailleurs été utilisée dans le domaine de l'animation d'humanoïdes [56]. L'algorithme CCD propose une méthode itérative pour amener l'effecteur sur la cible. Il se base sur l'alignement successif des segments avec la cible. Au fur et à mesure des itérations, l'effecteur va converger vers la cible (figure 3.3).

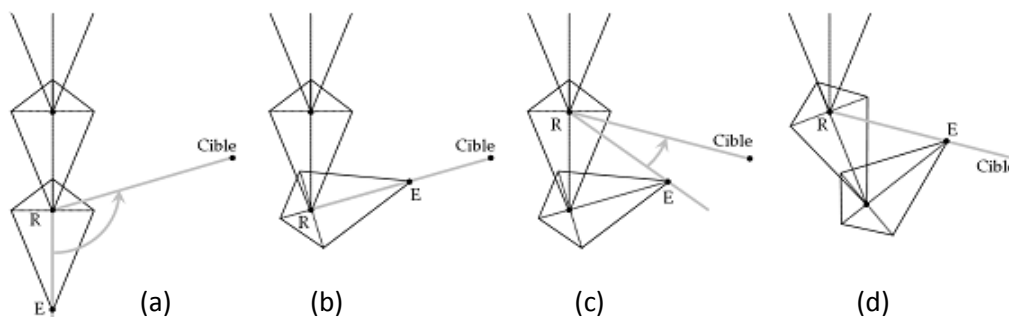


Figure 3.3 : Exemple d'utilisation du CCD. (a) et (b) Le dernier segment est adapté de manière à aligner R, E et Cible. (c) et (d) Le premier segment est adapté à son tour.

Malgré un calcul extrêmement rapide de chaque adaptation et le fait que cette méthode converge toujours sans singularité, elle reste très peu utilisée dans le domaine de l'animation d'humanoïdes. En effet, elle comporte de gros inconvénients :

- Il n'est pas possible d'ajouter des tâches secondaires. Cet algorithme ne fournit pas l'espace des solutions possibles en tenant compte des redondances mais uniquement une

solution. Il faut donc impérativement contrôler l'adaptation afin que l'algorithme converge directement vers la solution souhaitée.

- La convergence du CCD est très rapide au début puis très lente lorsque l'effecteur se rapproche de sa cible. La figure 3.3 montre l'itération qui suit celle présentée dans la figure 3.2. Contrairement à la première itération, l'extrémité ne s'est pas beaucoup rapprochée de la cible. Les itérations suivantes sont dessinées dans la dernière image à droite (d). La convergence ralentit lorsque l'effecteur se rapproche de la cible. C'est la raison pour laquelle Wang et coll. [57] ont utilisé le CCD uniquement pour se rapprocher très vite de la solution et a ensuite terminé la cinématique inverse avec une optimisation de type Broyden-Fletcher-Shanno [65].

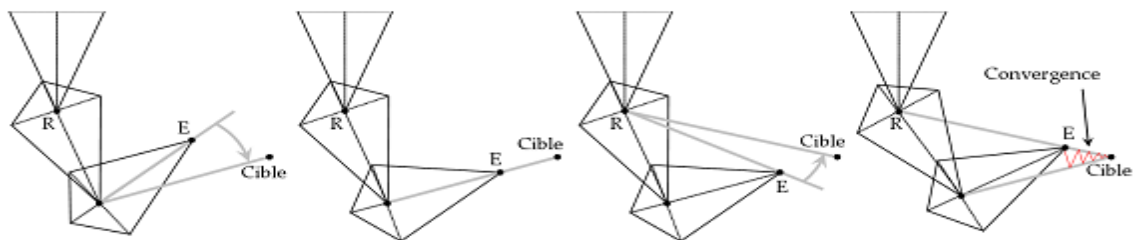


Figure 3.3 : Cette figure présente la seconde itération qui suit celle de la figure 3.2. L'image de droite montre les adaptations réalisées par les itérations suivantes (en rouge).

3.4.5 Méthode FABRIK (Forward And Backward Reaching Inverse Kinematics)

D'autres méthodes proposent de résoudre le problème de cinématique inverse (IK) en se plaçant, non plus dans l'espace des orientations mais dans celui des positions (Espace Euclidien). Andreas a proposé sa méthode appelé FABRIK [45] qui permet de résoudre différents problèmes de la cinématique inverse pour plusieurs cas de figure, que ça soit pour des systèmes simples ou bien au contraire pour des systèmes complexes avec une ou plusieurs cibles à atteindre. Elle a été introduite pour la modélisation et les suivis de la main. Andreas a comparé sa méthode avec d'autres méthodes de la cinématique inverse [45], il a trouvé qu'elle est plus rapide et plus fiable. Elle gère les limites angulaires ce qui rend les résultats plus au moins réalistes. La méthode FABRIK résout le problème de IK dans l'espace des positions au lieu de l'espace d'orientation. La solution est basée sur un déplacement en avant et en arrière jusqu'à l'obtention de la nouvelle position de l'articulation.

3.4.6 Bilan

La cinématique inverse est utilisée afin de contrôler les postures de figures articulées. Elle est appliquée dans plusieurs domaines tel que dans la robotique, l'animation par ordinateur, l'ergonomie et dans la production des jeux vidéo. Dans les sections précédentes nous avons présenté la plupart des travaux de la cinématique inverse. Le tableau 3.1 présente une comparaison entre les différentes méthodes de la cinématique inverse par rapport au temps de calcul, la simplicité et la continuité des résultats.

Méthode cinématique inverse	Simplicité	Rapidité	Autres caractéristiques
Analytiques	Simple	lent	-Solution n'existe pas toujours. - le système est limité à contrôler quelques membres.
Méthode itératives (Jacobien)	difficile	Très lent	-coûteuse en terme temps calcul -L'inverse n'existe pas toujours : solutions données par transposée jacobienne et le pseudo inverse.
Méthode basée sur le gradient	Simple	rapide	-Manque de réalisme. -Solution n'existe pas toujours.
CCD	Simple	rapide	-Mouvements peu naturels -Le nombre d'itération pour parvenir à la solution est généralement très élevé. -solution n'existe pas toujours (problème de convergence).
FABRIK	Très simple	Très rapide	-Nécessite des adaptations au cas des personnages. - Solution existe toujours.

Table 3.1 : Comparaison des principales méthodes de cinématique inverse.

3.5 Conclusion

Dans ce chapitre, Nous avons donné un aperçu sur les classes des méthodes de la cinématique inverse, ainsi que les travaux les plus importants dans chaque classe, ensuite, nous avons donnée un bilan qui représente une comparaison entre les plus principales méthodes utilisées dans la réalité virtuel, chaque méthode à ses avantages et ses inconvénient, le choix d'une méthode a utilisé est liée au système où nous allons utilisée cette méthode. Par exemple pour les applications interactives, il est souvent conseillé de choisir une méthode rapide.

V. Chapitre 4: Animation d'un humanoïde virtuel par la méthode FABRIK

4.1 Introduction

Dernièrement, les méthodes d'animation traditionnelles sont remplacées par des méthodes d'animation à base de mouvements déjà capturés. Ces méthodes sont capables de créer des mouvements d'une manière rapide sans beaucoup d'interaction de l'utilisateur, ce qui diminue le temps d'obtention des mouvements. Néanmoins, elles souffrent de certains inconvénients, le mouvement capturé n'est pas applicable pour un autre squelette, Il ne répond pas toujours aux exigences de l'utilisateur (mouvement difficile réaliser par l'acteur).

Donc, le mouvement capturé a besoin d'être modifié afin de satisfaire les besoins de l'utilisateur. Nous cherchons donc à introduire une technique facile pour les modifications des mouvements induisant des résultats naturels. La technique proposée permet d'appliquer le mouvement d'un sujet à d'autres personnages virtuels qui n'ont pas les mêmes caractéristiques. Par la suite, nous allons présenter notre méthode à base de la cinématique inverse (FABRIK) dans le but consiste à modifier une posture pour atteindre un cible.

4.2 Méthode FABRIK (Forward And Backward Reaching Inverse Kinematics)

La méthode FABRIK propose un nouvel algorithme de cinématique inverse qui permet de remonter de proche en proche aux différentes positions des articulations d'une chaîne, en parcourant la chaîne en avant et en arrière en un nombre fini d'itérations. L'intérêt principal de cette approche réside dans la possibilité de contrôler des systèmes multi- effecteurs mais également dans sa rapidité d'exécution [45], elle permet de résoudre différents problèmes de la cinématique inverse, pour des systèmes simples ou bien des systèmes complexes avec une ou plusieurs cibles à atteindre. Cette méthode nous permet d'avoir des postures assez convaincantes après les calculs, car on obtient des mouvements lisses sans discontinuités et avec beaucoup de réalisme. Les algorithmes présentés ci-après utilisant cette méthode, vont résoudre les différents cas de figure incluant plusieurs fins d'effecteurs avec plusieurs cibles et en présence de fortes restrictions dans les articulations.

a. Principe de fonctionnement

La méthode FABRIK commence du dernier jointure vers le premier jointure, en effectuant les modifications une après une, puis elle fait l'opération inverse de la base allant vers l'effecteur de fin, cette méthode à l'inverse des autres, se base sur le principe de trouver un point sur une ligne et non de calculer des angles. L'algorithme suivant [45] nous donne une explication en détail du mode de fonctionnement dans le cas le plus simple (c'est-à-dire une chaîne articulée avec une cible) et sans la présence de contraintes.

Algorithme 4.1 Itération complète de FABRIK [45]

Entrées : Les positions des joints P_i pour $i = 1, \dots, n$, La position de la cible t

Les distances entre les joints $d_i = |P_{i+1} - P_i|$ /for $i = 1, \dots, n-1$

Sorties : Les nouvelles positions des joints P_i pour $i = 1, \dots, n$.

```

dist =  $|P_1 - t|$  /* La distance entre la base et la cible */
/* Vérifie si la cible est atteignable */
Si  $dist > d_1 + d_2 + \dots + d_{n-1}$  alors /* La cible n'est pas atteignable */
    Pour  $i = 1, \dots, n-1$  faire /* Trouver la distance  $r_i$  entre la cible  $t$  et les positions  $P_i$  */
         $r_i = |t - P_i|$ 
         $\lambda_i = d_i / r_i$ 
         $P_{i+1} = (1 - \lambda_i) \cdot P_i + \lambda_i t$  /* Trouver les nouvelles positions  $P_i$  */
    Fin pour
Sinon /* La cible est atteignable; alors, définir  $b$  La position initiale de  $P_1$  */
     $b = P_1$ 
    /* Vérifier si la distance entre l'effecteur de fin  $P_n$  et la cible  $t$  est plus grande à une certaine
    tolérance */
     $difA = |P_n - t|$ 
    Tant que  $difA > tol$  faire
        /* ETAPE 1 : Déroulement vers l'avant (FORWARD REACHING) */
         $P_n = t$  /* fixer l'effecteur de fin comme étant la cible */
        Pour  $i = n-1, \dots, 1$  faire
            /* Trouver la distance  $r_i$  entre les nouvelles positions  $P_{i+1}$  et les positions  $P_i$  */
             $r_i = |P_{i+1} - P_i|$ 
             $\lambda_i = d_i / r_i$ 
             $P_i = (1 - \lambda_i) \cdot P_{i+1} + \lambda_i P_i$  /* Trouver les nouvelles positions  $P_i$  */
        fin pour
        /* ETAPE 2 : Déroulement vers l'arrière (BACKWARD REACHING) */
         $P_1 = b$  /* fixer  $P_1$  comme position initial */
        Pour  $i = 1, \dots, n-1$  faire
            /* Trouver la distance  $r_i$  entre les nouvelles positions  $P_i$  et les positions  $P_{i+1}$  */
             $r_i = |P_{i+1} - P_i|$ 
             $\lambda_i = d_i / r_i$ 
            /* Trouver les nouvelles positions  $P_i$  */
             $P_{i+1} = (1 - \lambda_i) \cdot P_i + \lambda_i P_{i+1}$ 
        fin pour
         $difA = |P_n - t|$ 
    fin tant que
fin si
fin
    
```

Explications l'Algorithme 4.1

On a comme entrées:

- P_i : les jointures de positions de la chaîne articulée avec $i = 1, \dots, n$. P_1 : la jointure de base et P_n : l'effecteur de fin, t : la cible qu'on veut atteindre, b : la position de base.

Les sorties sont représentées comme les nouvelles positions des jointures de P_i^f avec $i = 1, \dots, n$.

Avant tout on va calculer les distances d_i entre chacun des jointures avec $d_i = |P_{i+1} - P_i|$, pour $i = 1, \dots, n - 1$. Afin de savoir si la cible t est atteignable ou pas, on calcule la distance entre la jointure de base et la cible t , $dist = |P_1 - t|$, et si la distance est inférieure à la somme totale des distances d_i , $dist < \sum_{i=1}^{n-1} (d_i)$ alors la cible est atteignable, dans le cas contraire. N'est pas atteignable et l'algorithme s'arrête.

Si la cible est atteignable alors une itération complète est lancée, cette itération se déroule en deux étapes. [45]

1- La première étape consiste à déterminer les positions de tous les jointures en commençant par l'effecteur de fin P_n , jusqu'au jointure de base P_1 , en premier lieu l'effecteur de fin va devenir la cible t , c'est-à-dire qu'on va projeter temporairement P_n sur la position de la cible t , $P_n^t = t$. Après on va tracer une ligne l_{n-1} , qui va passer par P_n^t et P_{n-1} et finalement P_{n-1}^t va être sur cette ligne avec une distance d_{n-1} de P_n^t .

2- La même opération est répétée pour les autres jointures jusqu'à arriver à la base P_1 . Prend bien une nouvelle position P_1^t , sachant que la position de base $b = P_1$ rester la même (elle est fixe), elle doit donc revenir à sa position initiale et pour ce la, il faut procéder à la deuxième étape de l'algorithme.

3- Dans cette deuxième étape, on procède de la même manière que l'opération précédente mais en commençant de la base en allant vers l'effecteur de fin, c'est-à-dire qu'on va fixer $P_1^t = b$ (la position de base initiale) donc P_1^t va être projeté vers b et on va obtenir la position P_1'' puis une ligne l_1 va être tracé et va passer de P_1'' vers P_2^t alors la nouvelle position de P_2^t qui est P_2'' va se mettre sur cette ligne l_1 d'une distance d_1 de P_1'' . Ainsi une itération complète s'est achevée.

Après avoir obtenu une itération complète, on vérifie si l'effecteur de fin a atteint sa cible, qu'il soit exactement sur elle ou suffisamment proche (selon des critères prédéfinie),

et si ce n'est pas le cas alors d'autres itérations ce mettront en marche afin d'atteindre la cible.

Généralement, la méthode FABRIK pour un système sans contraintes, converge toujours vers un résultat, tant que la somme de la longueur des chaînes est supérieure à la distance entre la jointure de base et la cible. Dans le cas contraire où la cible n'est pas atteignable, il y'a quand même un critère de fin pour l'algorithme, ce critère de fin peut être définie par une comparaison entre la position initiale et actuelle de l'effecteur de fin et cette distance entre les deux positions doit être inférieur à une certaine tolérance prédéfinie. Dans le cas extrême où la cible n'a pas pu être atteinte et où le nombre d'itérations devient conséquent, alors on fixe un nombre d'itération à ne pas excéder et l'algorithme est brusquement arrêté.

b. Itération complète de FABRIK (exemple)

La figure (4.1) suivante résume le fonctionnement de l'algorithme (4.1) :

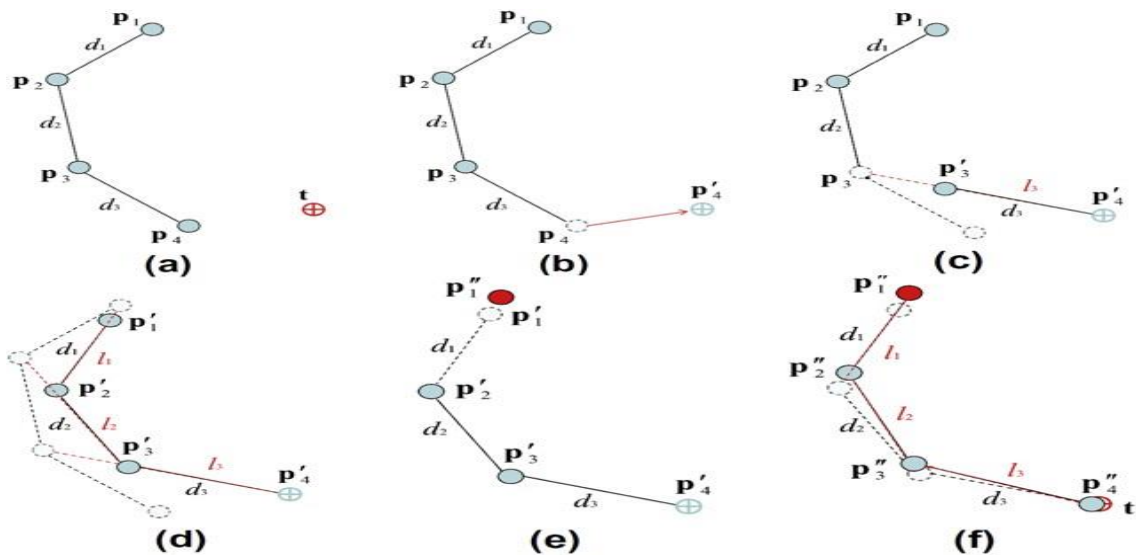


Figure 4.1 : Itération complète de FABRIK

Dans cet exemple on a une chaîne articulée qui comporte 4 jointures ($P_i = P_1, \dots, P_n$) et où l'effecteur de fin doit atteindre une cible t .

- a) La position initiale de la chaîne et la cible t .
- b) Déplacement de P_4 vers la cible t .
- c) La nouvelle position de P_3 est sur la ligne l_3 qui passe par P_4' et P_3 et qui est d'une

distance d_3 de P'_4 .

- d) L'opération est répétée pour tous les jointures.
- e) La deuxième étape de l'algorithme commence : déplacement de P'_1 et P''_1 et qui est la position initiale b.
- f) L'opération est répétée mais cette fois ci du sens inverse. Il est important qu'une méthode de cinématique inverse puisse résoudre un cas plus complexe et qui reflète la réalité comme des parties du corps humain comme une main ou une épaule, qui peuvent se composées de plusieurs effecteurs de fin et/ou de plusieurs cibles à atteindre et aussi de plusieurs sous bases qui relient plusieurs chaines entre elles.

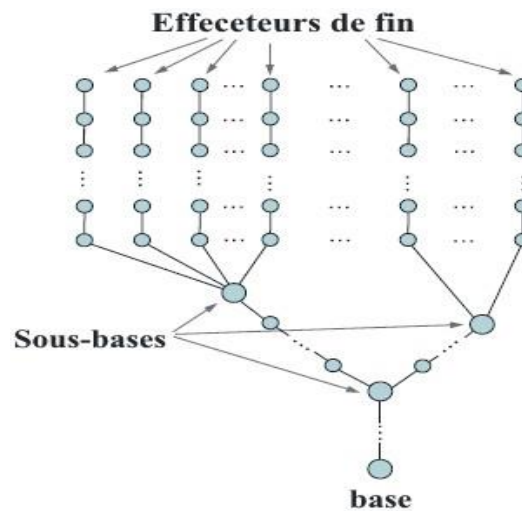


Figure 4.2: Chaîne avec plusieurs sous-bases.

Le principe de FABRIK pour un cas complexe est pratiquement le même que pour un cas simple c'est-à-dire que pour un système composé de plusieurs chaines reliées à une sous base, l'algorithme va appliquer la première étape sur chacune des chaines, des effecteurs de fin vers leur base parente et qui est une sous base partageant avec d'autres sous base une autre base parente et il va continuer a traité ainsi autant de sous bases intermédiaires jusqu'à arriver à la base principale, alors la deuxième étape pourra enfin commencer. Elle consiste à faire l'opération inverse, traitant toutes les chaines séparément en allant de la base principale vers les sous bases, puis des sous bases vers les effecteurs de fin qui pourront atteindre leur cibles, le nombre d'opérations est répété autant de fois que nécessaire, on peut avoir donc plus d'une itération.

c. FABRIK avec contraintes

La plupart des parties du corps humain présentent bien visiblement des restrictions biomécaniques naturelles, donc pour la représentation de modèles de certaines parties du corps humain, il est primordial d'introduire des contraintes pour obtenir un certain réalisme. Beaucoup de paramètres entrent en jeu afin de pouvoir représenter au mieux un modèle correcte, ces paramètres vont déterminer l'espace de mouvement et la structure du modèle. Les jointures sont les principaux paramètres qui décrivent la structure d'un modèle, par exemple le modèle de la colonne vertébrale qui est représentée par 24 jointures ou l'épaule qui comporte 3 jointures.

d. Contrainte sur la jointure

Un jointure est définie dans l'espace de mouvement par sa position et son orientation qui sont possible par son nombre de DDLs, pour ce qui suit on va se concentrer sur le cas le plus général où la jointure possède 3 DDLs, 2 degrés de rotation simple qui représentent la possibilité à la chaîne d'atteindre le vecteur de destination et 1 degré de liberté qu'on appelle d'orientation qui va faire une rotation autour de ce vecteur. Donc ces 3 DDLs vont représenter les restrictions sur les jointures Prenons comme exemple de jointure, une rotule avec une limite d'orientation représentée par le rotor R , et une limite de rotation représentée par $\theta_1, \theta_2, \theta_3$ et θ_4 . Alors on obtient un cône qui va décrire les restrictions de rotation sur la jointure comme illustré dans la figure (4.3) suivante :

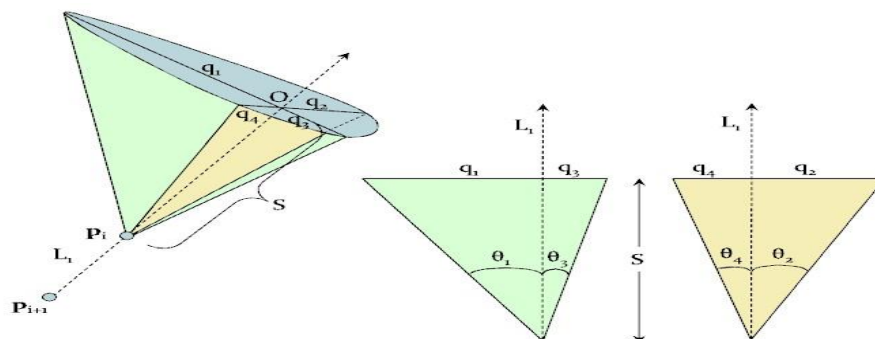


Figure 4.3: Cône représentant Les restrictions de rotations.

Avec S : Distance entre la jointure et le centre de l'ellipse.

$q_j : j = 1, \dots, 4$. avec $q_j = S.\tan(\theta_i)$ et le champ de mouvement est représenté en bleu. On peut avoir 3 différentes sortes de représentation conique selon les angles θ_i comme montrer sur la figure ci-dessous :

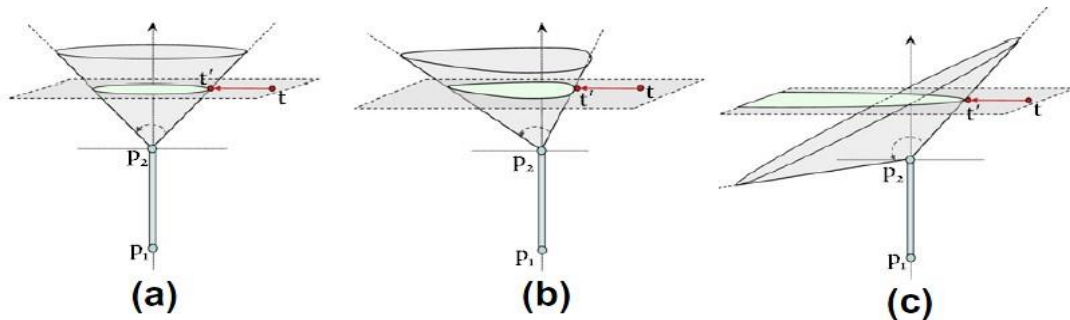


Figure 4.4: Différentes sortes de cônes.

- a): Les θ_i sont identiques alors on obtient un cercle,
- b): Les θ_i sont différents alors on obtient une forme ellipsoïdale.
- c): Deux θ_i seulement sont identiques et sont supérieur ou inférieur à 90° alors on obtient une parabole.

e. Algorithme pour les restrictions d'orientation

Imaginons qu'on soit dans la première étape de l'algorithme (4.1) citée précédemment, précisément à l'instant où on a trouvé la nouvelle position du jointure $(i-1)^{ième}$, la démarche de l'algorithme pour une restriction d'orientation est décrite comme suit [45] :

Algorithme 4.2 Restrictions sur l'orientation

Entrées : La rotation R qui exprime les restrictions d'orientation sur P_i

Sorties : La nouvelle orientation de P_i et qui est P'_i .

Si La rotation est dans la zone possible de mouvement **Alors**
ne rien faire et quitter

Sinon

Réorienter la jointure P'_i de façon a ce que la rotation R sois dans le dans le champ de mouvement.

Fin si

Fin

On va d'abord détecter la restriction d'orientation appliquée sur le jointure, de là on va vérifier l'orientation du jointure, si elle est en dehors des limites possible du mouvement, on va réorienter la jointure de façon à rester dans les limites prédéfinies. Comme illustré sur la figure 4.5:

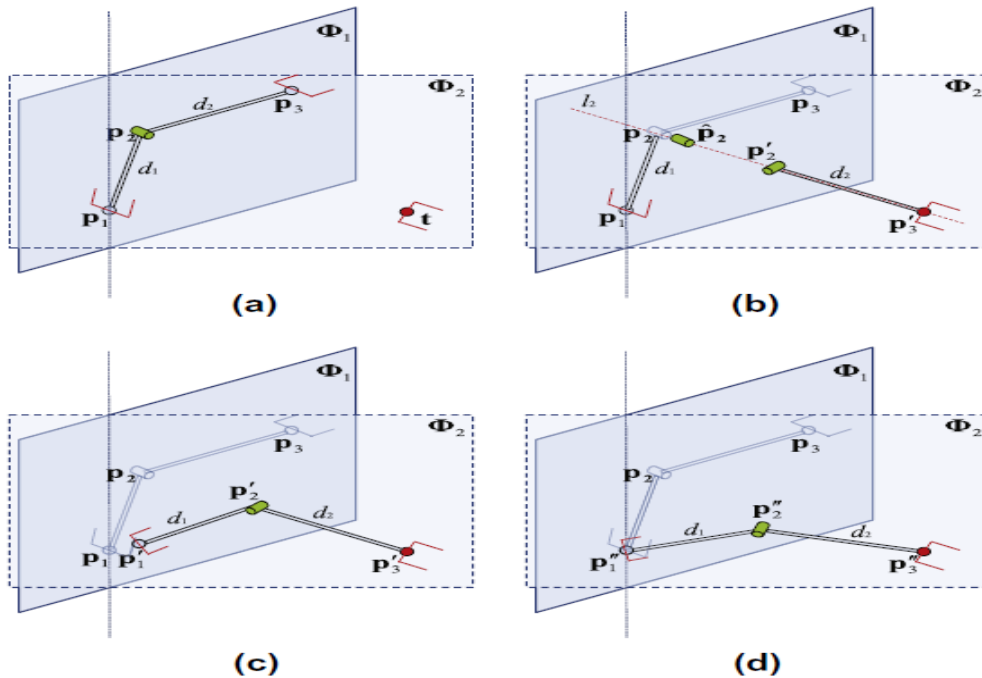


Figure 4.5: Contraintes d'orientation

(a) La position initiale du bras manipulateur de la cible t . Le plan Φ_1 représente le plan d'orientation pour P_2 (un degré de liberté), le plan Φ_2 représente la position de la cible.

(b) Projection et orientation de P_3 sur la cible t , puis projection de P_2 sur Φ_2 obtenant \hat{P}_2 après, tracer la ligne l_2 qui passe par P_3 et \hat{P}_2 et sur laquelle va se positionner P'_2 à une distance d_2 de P_3 . Réorienter P'_2 en appliquant les contraintes d'orientations.

(c) Trouver P'_1 qui se trouve sur la ligne l_1 qui passe par P_1 et P'_2 , avec une distance d_1 de P_1 .

(d) Maintenant que tout est sur le plan Φ_2 donc le problème devient un problème en 2D, alors FABRIK peut être utilisé dans sa forme la plus simple.

f. Algorithmes pour les restrictions de rotation

On va expliquer la méthode Fabrik avec des restrictions de rotation en deux parties.

F.1) Déroulement vers l'avant (Forward reaching).

Algorithme 4.3 Déroulement vers l'avant avec restrictions de rotation

Entrées : La position actuelle de la cible t , les θ_i qui déterminent le cône, d_i les distances entre P_i et P_{i-1} .

Sorties : Les nouvelles positions des joints p_i pour $i = 1, \dots, n$.

Tracer la ligne l_1
 Trouver O qui est la projection de la cible sur la ligne l_1
 /* O est l'origine des axes y et $x \Rightarrow$ passage au plan simplifier en 2D */
 Trouver les nouvelles positions de P_i , notées \hat{P}_i .
Si \hat{P}_i est dans le champ de mouvement **Alors**
 Ne rien faire
Sinon
 Projeter \hat{P}_i sur l'ellipse et voir dans quel quadrant est le point plus proche, \hat{P}_i
 Translater \hat{P}_i sur la ligne du cône d'une distance d_i .
 P_i la nouvelle position de P_i
Fin si
Fin

On a comme données :

La position de la cible t que la quelle on va projeter la jointure P_i . La distance d_{i-2} qui est la distance entre la jointure P_{i-1} et P_{i-2} . On suppose des restrictions de rotation entre les jointures P_i et P_{i-1} , qui sont représentés par un cône limité pas les angles θ_i . Dans un premier temps On va projeter la cible t sur la ligne l_1 qui passe par P_i et P_{i-1} , et sur cette ligne ; d'une distance S de P_{i-1} , on obtient l'axe déterminant le cône. Enfaite ce cône qui a comme origine la jointure P_{i-1} , est la restriction de mouvement sur la jointure P_{i-2} .

On va vérifier que P_{i-2} est dans le champ de mouvement possible si ce n'est pas le cas et qu'il est en dehors du cône, alors on va fixer P_{i-2} sur le point le plus proche du cône et on va le translater afin qu'il soit d'une distance d_{i-2} du jointure P_{i-1} . On va suivre la même procédure pour la suite.

F.2) Déroulement vers l'arrière (Backward reaching).

Ici, on fait l'opération inverse (voir algorithme 4.1) jointure par jointure jusqu'à arriver à la cible en vérifiant toujours les contraintes de rotation qu'on va anticiper de la même manière que dans l'algorithme (4.3).

Si il y'a des restrictions de rotation sur le dernier jointure et qui ne peut pas atteindre la cible alors on projette la cible sur le point le plus proche du cône comme montrer dans la figure ci-dessous :

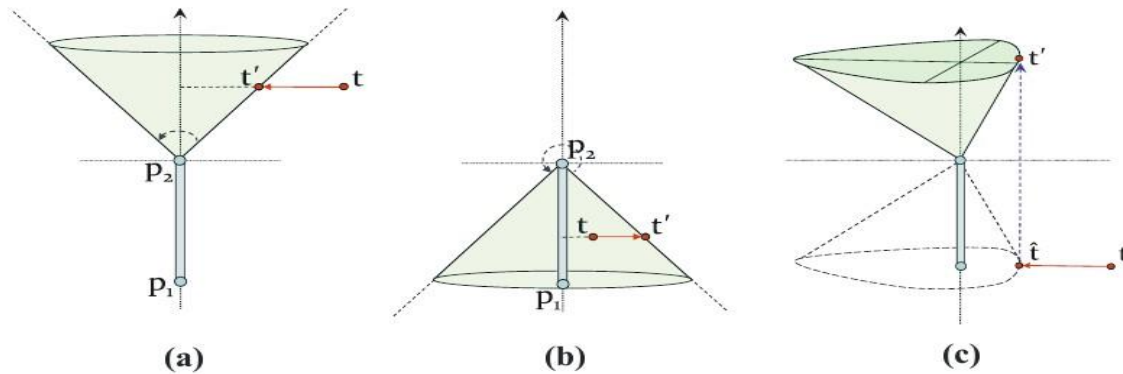


Figure 4.6: Les possibilités de positionnement pour t.

D'après la figure, On distingue trois cas possibles :

- a) La cible est projetée sur le point le plus proche du cône.
- b) Le plan de restriction dépasse les 180 degrés.
- c) La cible est localisée dans un hémisphère différent du cône.

Et pour finir on refait la même opération autant de fois que nécessaire jusqu'à atteindre la cible (convergence). L'algorithme (4.4) suivant résume ce qui a été dit plutôt :

Algorithme 4.4 : Déroulement vers l'arrière avec restrictions de rotation

Entrées : La position de base b, di les distances entre P_i et P_{i-1} .

Sorties : La nouvelle position temporaire de t

```

Si il y'a des restrictions de rotation sur  $P_i$  alors
  Si la cible t est atteignable alors
    ne rien faire
  Sinon
    Projeter la cible sur le point le plus proche du cône
  Fin si
Fin si
Fin
    
```

Exemple

Voici la figure 4.7 présente un exemple d'application de FABRIK sur un bras qui comporte des contraintes d'orientation et de rotation.

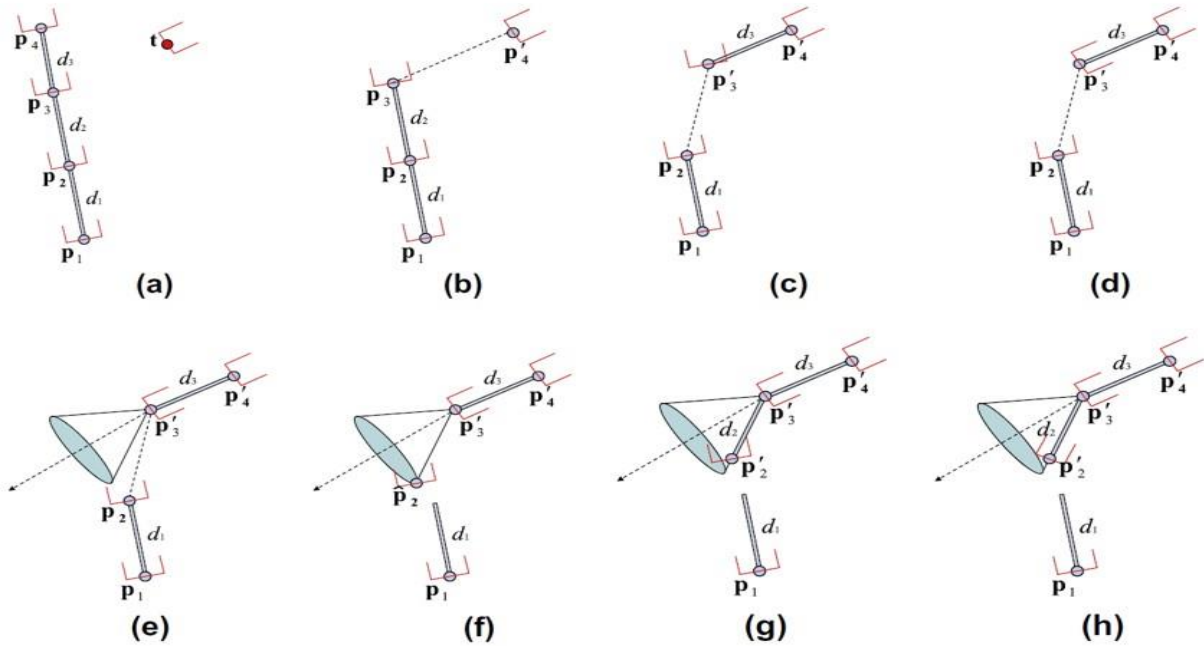


Figure 4.7: FABRIK avec des contraintes d'orientation et de rotation

- a) Configuration initiale de la chaîne et de la cible t .
- b) Relocaliser et réorienter P_4 sur la cible t .
- c) Déplacer la jointure P_3 à P'_3 sur la ligne qui passe par les points P'_4 et P_3 et d'une distance d_3 de P'_4 .
- d) Réorienter la jointure P'_3 de façon à ce que le rotor exprimant possibilités d'orientations entre P'_3 et P'_4 , soit conforme aux limites prédéfinies.
- e) Les limites de rotations sont représentées par un cône qui a comme axe, la ligne qui passe par les points P'_4 et P_3 .
- f) Relocaliser P_2 en \hat{P}_2 , \hat{P}_2 est le point le plus proche appartenant au cône, ainsi \hat{P}_2 est dans les limites du champ de mouvement défini par les contraintes de rotations.
- g) Déplacer \hat{P}_2 à P'_2 pour respecter la longueur du lien.
- h) Réorienter P'_2 afin de satisfaire les contraintes d'orientation.

4.3 Technique d'animation d'un humanoïde virtuel par FABRIK

Nous allons proposer dans ce qui suit la réalisation de la technique FABRIK pour l'animation d'un humanoïde virtuel pour atteindre cet objectif nous allons effectuer un travail à l'aide des deux modules : le premier pour lire le fichier BVH et lier le squelette défini dans le BVH au modèle (Humanoïde) choisi. La matrice BVH utilisée pour animer l'humanoïde,

Le deuxième module, utilise la cinématique inverse afin d'atteindre une cible, éviter un obstacle ou ajouter un mouvement spécifique (toucher la tête par la main à gauche). Nous avons intégré la méthode FABRIK sur l'humanoïde (notre modèle). Le schéma général de l'application proposée est représenté dans la figure suivante (figure 4.8).



Figure 4.8: Schéma proposé pour gérer l'application.

L'application proposée reçoit comme entrée un fichier de capture de mouvements avec l'extension BVH et l'humanoïde modélisé, et produit comme sortie un humanoïde en mouvement. Le schéma détaillé de la technique est donné par la figure 4.9.

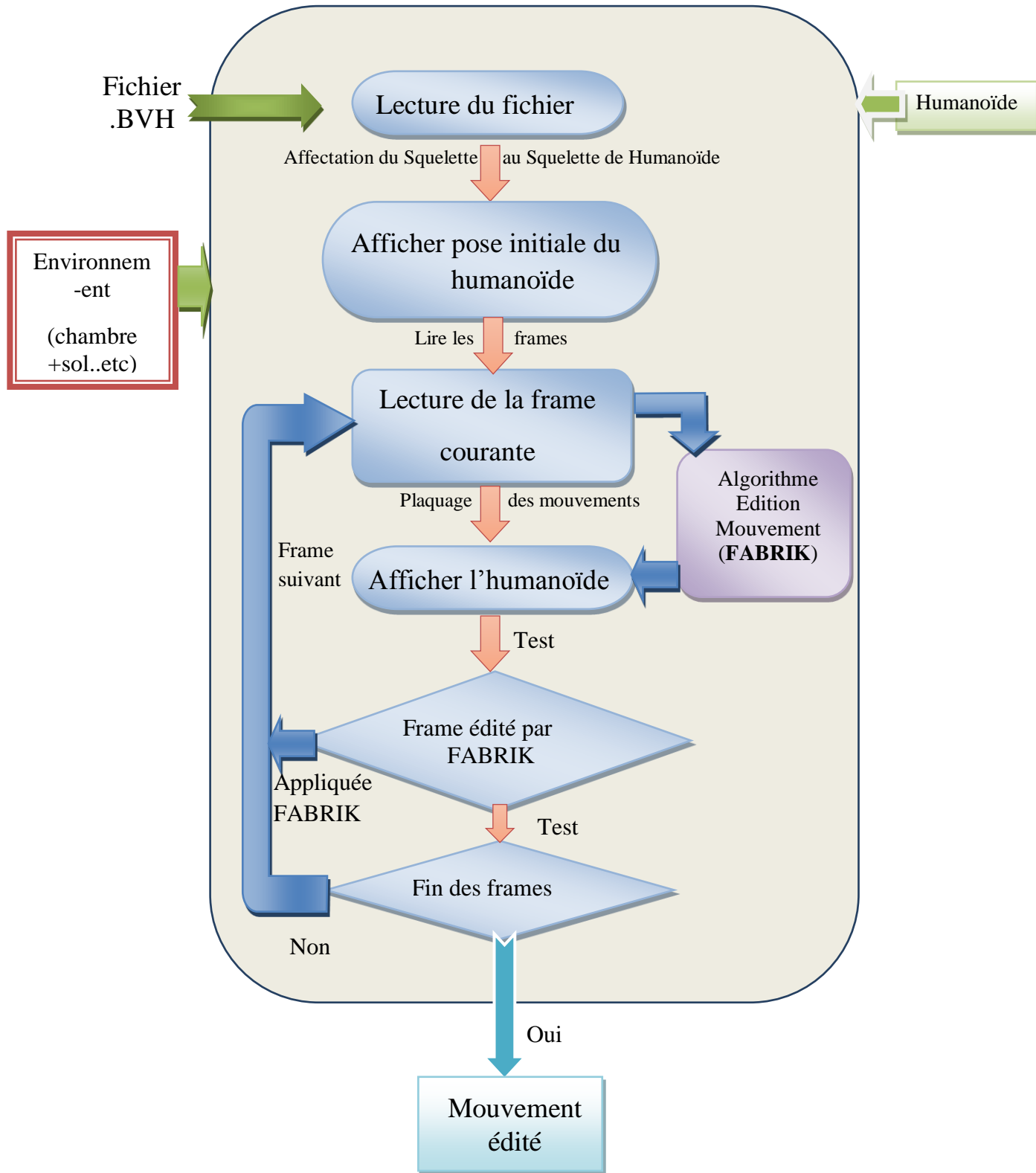


Figure 4.9: Schéma détaillé de la technique.

4.3.1 Module 1 : Animation par capture des mouvements

a) Lecture le fichier BVH

La première étape de ce module consiste à faire l'extraction du squelette ayant la même structure que celle définit dans le fichier BVH et l'extraction des paramètres du mouvement qui existe dans la partie entête du fichier BVH par exemple (nombre de Frame, time Frame...etc.), puis extraction des données 3D des transformations (partie mouvement). Nous allons utiliser ces informations pour la création de la matrice BVH, voici un schéma qui illustre ce module.

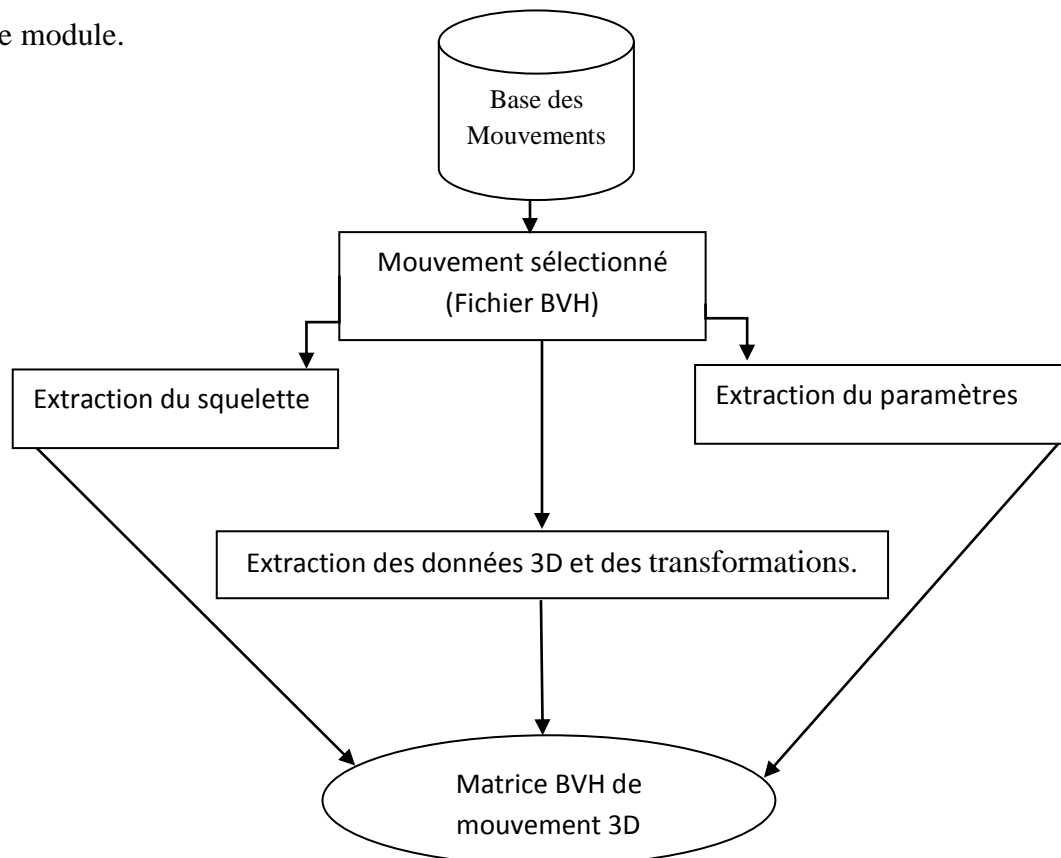


Figure 4.10: lecture fichier BVH.

a.1) Structures des données

Structure Jointure

```

|   Nom: char [20];
|   Index : entier;
|   Parent : pointeur vers la même structure;
|   Fils : liste de type Jointure;
|   Offset : tableau de trois entiers; //position de la jointure
|   A_un_site : boolien; //si la fin de la branche ;
|   Channel : liste de type Channel ;
|   Distance : distance entre jointure et jointure fils ;

```

Fin.

Structure Channel

| Type : ChannelEnum ;
| Index : entier ;

Fin.

Énumération ChannelEnum

| X_ROTATION, Y_ROTATION, Z_ROTATION,
| X_POSITION, Y_POSITION, Z_POSITION

Fin.

Algorithme 4.5 : Algorithme Lecture du fichier (*.BVH)

Ouvrir le fichier (*.BVH).

TQ n'est pas fin du fichier **Faire**

Prendre une ligne.

Eviter les séparateurs comme : , \t .

Si '{' **alors**

Ajouter à la fin de la liste des jointures. **Fin si**

Si '}' **alors**

retour arrière de la liste des jointures. **Fin si**

Si 'ROOT' ou 'JOINT' **alors**

Nouvelle jointure.

Remplissage du nom et parent. **Fin si**

Si 'CHANNELS' **alors**

Remplissage de degrés de liberté (Channel). **Fin si**

Si 'OFFSET' **alors**

Remplissage de la position de la jointure. **Fin si**

Si 'END' **alors**

La fin de la branche.

Construction alors de la liste des fils. **Fin si**

Si 'MOTION' **alors** /*la partie de mouvements.*/

Prendre une ligne. **Fin si**

Si 'Frames' **alors**

lire le nombre de frames. **Fin si**

Si 'Frame Time' **alors**

lire le temps de chaque frame.

/* Construction du tableau du mouvement. */

Fin si

Taille = nombre de frame * nombre de Channel.

Pour i=0 à nombre de frame **Faire**

Prendre une ligne et éviter les séparateurs.

Pour j=0 à nombre de Channel **Faire**

Prendre une ligne et éviter les séparateurs.

Tableau_mouvement [i*nombre de Channel +j] ;

j←j+1 ;

Fin Pour ;

i← i+1 ;

Fin Pour ;

Fin TQ ;

Fermeture de fichier.

Fin

b) Relier le squelette de BVH avec le squelette de l'Humanoïde (mapping) ou Plaquage des mouvements

Cette étape consiste à relier chaque jointure extrait du fichier capture mouvement à chaque jointure du modèle (Humanoïde) pour l'adaptation du mouvement. A réaliser une étape très importante et assez difficile. On peut simplifier l'opération en utilisant le tableau suivant pour définir plusieurs jointures qui relient chaque jointure de fichier capture mouvement avec une jointure du squelette d'humanoïde (notre modèle).

Type	Membre	Jointure 1	Jointure 2	Jointure 3	Jointure 4	Jointure 5	Jointure 6
Squelette BVH modèle 1	ped gauche	Hip	Left Up Leg	Left Leg	Left Foot	Left Toe Base	
	ped a droit	Hip	Right Up Leg	Right Leg	Right Foot	Right Toe Base	
	Tête	Lower Back	Spine	Spine1	Neck	Neck1	Head
	main gauche	Left Shoulder	Left Arm	Left Fore Arm	Left Hand	Left Finger Base	Thumb
	main a droit	Right	Right Arm	Right Fore Arm	Right Hand	Right Finger Base	Thumb
Squelette BVH modèle 2	ped gauche	Hips_1	Left Up	Left Up Leg_1	Left Leg_1	LeftFoot_1	Left Toe Base
	ped a droit	Hips_1	Right Up	Right Up Leg_1	Right Leg_1	Right Foot_1	Right Toe Base
	Tête	Spine_1	Spine1	Spine3	Neck_1	Head_1	Head End
	main gauche	Left Shoulder_1	Left Arm_1	Left Fore Arm_1	Left Hand_1	Left Hand Index1	Left Hand Index1
	main a droit	Right Shoulder	Right Arm	Right Fore Arm	Right Hand	Right Hand Index1	Right Hand Thumb1
Squelette Humanoïde (notre modèle)	ped gauche	Waist(Root)	LHip	LKnee	Lankle		
	ped a droit	Waist(Root)	RHip	RKnee	Rankle		
	Tête	Sterno-Clavicular					
	main gauche	Sterno-Clavicular	LShoulder	LElbow	LWrist		
	main a droit	Sterno-Clavicular	RShoulder	RElbow	RWrist		

Tableau 4.1 : Définition de la hiérarchie des squelettes avec le nom des membres et des jointures pour chaque modèle utilisé BVH1, BVH2 et notre Humanoïde.

c) Utilisation de la matrice BVH pour animer l'humanoïde

Les matrices BVH pour chaque jointure sont représentées comme suit :

$$\text{Matrice BVH} = \begin{pmatrix} \cos(a) & -\sin(a) & 0 & \text{offset } x \\ \sin(a) & \cos(a) & 0 & \text{offset } y \\ 0 & 0 & 1 & \text{offset } z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La colonne 4 de la matrice BVH définit l'offset de la jointure par rapport au père et a l'angle des données 3D des transformations.

Pour animer notre humanoïde par matrice BVH, on utilise la méthode d'affichage suivante :

Algorithme 4.6 : Algorithme Affichage récursive Humanoïde

Entrées : Jointure : pointeur vers la structure Jointure , Taille

Sorties : Série des poses d'humanoïde.

Animer_Humanoïde (jointure, donnée, taille, numéro_frame)

Début

Si (jointure.nom = la racine) **alors**

 Translation par rapport au repère scène avec le vecteur (jointure.Offset*taille)

Sinon

 Translation par rapport au père avec le vecteur (jointure.Offset * taille)

Fin si

 /* choisir le mode de rotation ZYX ou ZXY selon le fichier BVH */

Si (jointure.Channel [1]=Z_ROTATION et jointure.Channel [2]=Y_ROTATION et jointure.Channel [3]=X_ROTATION) **alors** R=1

Sinon R=2

Fin si

 /* orientation os relier par jointure ont utilisé la méthode **EulerToQuat** */

Orientation (EulerToQuat (Tableau_mouvement [numéro_frame * jointure.channel.index], Tableau_mouvement [numéro_frame * jointure.channel.index+1], Tableau_mouvement [numéro_frame * jointure.channel.index+2], R));

 /* appliqué pour tous jointure fils */

Pour i=0 à jointure.fils **Faire**

 Animer_Humanoïde (jointure.fils[i], donnée, taille)

 i ← i+1 ;

Fin Pour

Fin ;

```

Quaternion EulerToQuat (vecteur angle_vecteur, R : entier);
  Début
    Matrice 3*3 mx; /* utiliser la matrice BVH */
    Si R=1 alors mx.FromEulerAnglesZYX (Degree(angle_vecteur.x),
      Degree(angle_Vecteur.y), Degree(angle_vecteur.z));
    Sinon mx.FromEulerAnglesZYX (Degree(angle_vecteur.x),
      Degree(angle_Vecteur.y), Degree(angle_vecteur.z));
    Fin si
    Quaternion (mx);
    return mx;
  Fin;
Fin;

```

Si numéro_frame =0 Afficher pose initiale du humanoïde, Les matrices BVH calculées et interpolées en utilisant une représentation par Quaternions et par angles d'Euler.

- **Les Angles d'Euler** sont décrits par les trois rotations successives autour de chacun des axes X, Y et Z. Cependant, des variantes existent selon l'ordre des rotations utilisées. Cette représentation très simple et intuitive pose néanmoins un problème de singularité connu sous le nom de "Gimbal Lock". Cette singularité se traduit par la perte d'un degré de liberté après une séquence de rotations d'angle $\pi/2$.

- **Les Quaternions** proposés par Lord Williams Hamilton [70] au 19^{ème} siècle sont une extension des nombres complexes puisque la rotation est symbolisée par $Q=w+xi+yj+zk$ où w, x, y et z sont des nombres réels et i, j et k sont trois axes orthonormés imaginaires tels que $ii =jj =kk=ijk =-1$. Les quaternions sont très utilisés en animation, parce qu'ils permettent une interpolation qui assure la prise du plus court chemin, pour aller d'une rotation à une autre.

4.3.2 Module 2 : utilisé la cinématique inverse afin d'atteindre une cible (intégré la méthode FABRIK sur l'humanoïde)

a) Modèle humanoïde

La modélisation d'un humanoïde virtuel dépend du mouvement à simuler, où on met plus de détails sur la partie du corps qui fait le mouvement (exemple : la main si le mouvement est la préhension), ainsi que sur le but de la simulation : Jeux, robotique, biomécanique, médecine.

La structure de notre modèle d'humanoïde se compose de chaîne cinématique rigide (les os ne se déforment pas ils sont rigides), reliés par des jointures angulaires. Les jointures rotationnelles possèdent différents degrés de liberté. Chaque os est au moins attaché à une jointure, ainsi une rotation d'une jointure entraîne la rotation de l'os attaché.

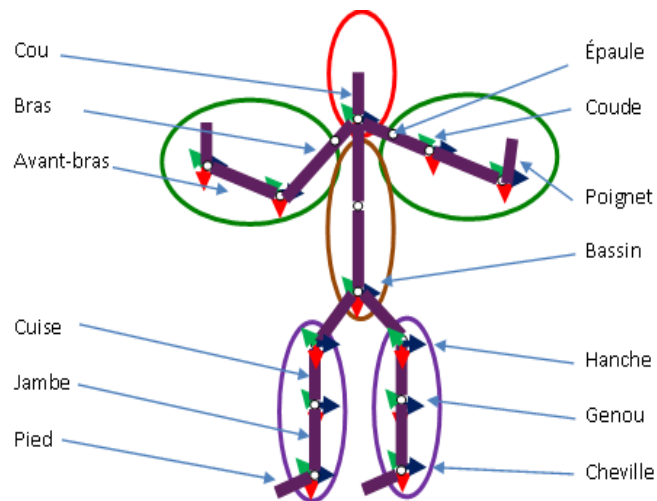
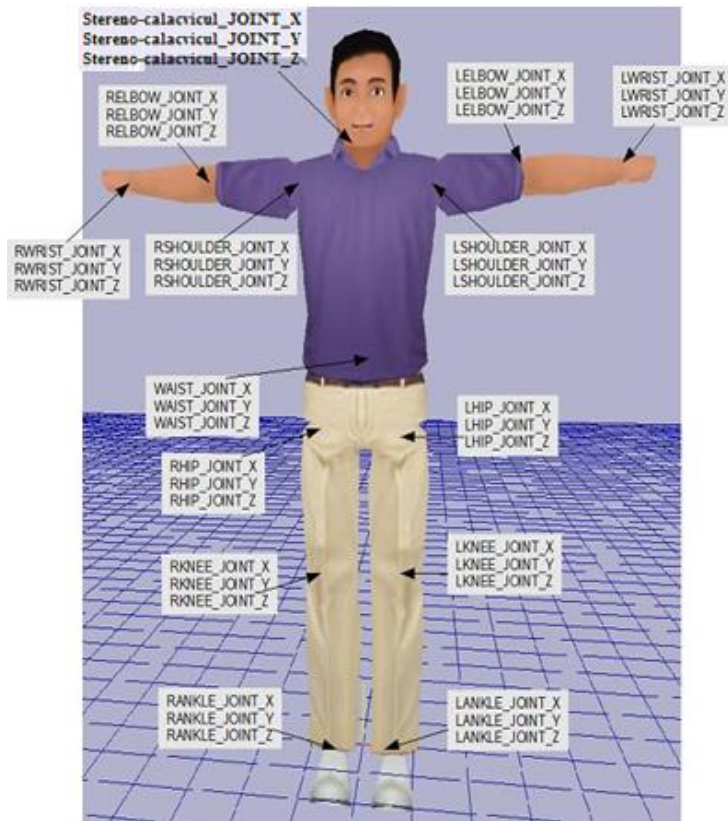


Figure 4.11 : Modèle d'humanoïde: hiérarchie proposée

Notre personnage virtuel est constitué de 22 articulations, 20 segments et avec 22 DDL. Les degrés de liberté sont réparties comme suit : le bassin (pelvis) à 6 degrés de liberté, 3 degrés de liberté pour translation et 3 degré de liberté d'orientation. Les articulations de coude, sterno-claviculaire, le poignet, genou et le cheville ont deux degrés de liberté. Les articulations de la hanche et l'épaule ont trois degrés de liberté. Figures 4.12,



Joint	DLL
Root ou Wa ist	6
Sterno-claviculaire	2
Shoulder	3
Elbow	2
Wirst	2
Hip	3
Knee	2
ank le	2

Figure 4.12 : Modèle d'humanoïde: Degré de liberté

L'humanoïde est divisé en groupes (en général la tête, le tronc, les bras (02 à gauche et à droite) et les jambes (02 à gauche et à droite)), les groupes sont reliés par des jointures "sous-bases" (voir **Figure 4.2**). Cette subdivision permet à notre humanoïde d'être très rapide en raison de l'application de la cinématique inverse (Méthode FABRIK) sur uniquement quelques articulations (nous appliquons seulement les articulations du groupe concerné par le mouvement).

a.1) Définition les jointures utilisées pour l'animation :

- a) **Jointure Root :** La racine (le pelvis du squelette) définit l'orientation de notre humanoïde dans la scène.
- b) **Jointure hip :** (La hanche) définit la rotation et translation du groupe jambe.
- c) **Jointure knee :** (le genou) et **la jointure ankle** (la cheville) utilisés pour l'orientation de partie inférieure de la groupe jambe.
- d) **Jointure Sterno-Clavicular :** (Le Sterno-claviculaire) définit l'orientation de groupe tête et le tronc.
- e) **Jointure shoulder :** (L'épaule) définit l'orientation de groupe bras,
- f) **Jointure elbow :** (Le coude) et **la jointure wrist** (Le poignet) utilisés pour l'orientation de partie supérieure de la groupe de bras.

a.2) Les chaines et Les effecteurs de fin

Le principe de notre modèle (humanoïde) composé des six chaines sont reliées par deux articulations (sous base):

- 1^{ere} chaine** : la tête,
- 2^{eme} chaine** : le bras à droite,
- 3^{eme} chaine** : le bras à gauche.
- 4^{eme} chaine** : la jambe à droite,
- 5^{eme} chaine** : la jambe à gauche,
- 6^{eme} chaine** : le tronc.

Les effecteurs de fin choisie de notre module (humanoïde) sont comme suit : **jointure Lankle, Rankle, Lwrist, Rwrist.**

Les sous bases : **Jointure Root et Jointure Sterno-Clavicular.**

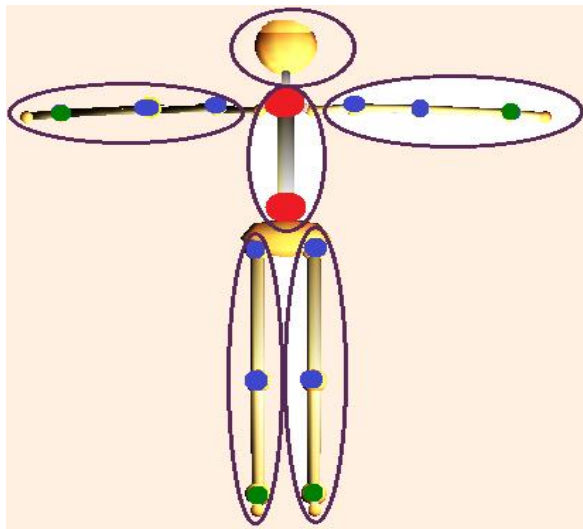


Figure 4.13: le point rouge désigne les sous base, le point vert indique les effecteurs finaux, le point bleu indique les jointures.

b) Edition de mouvement basée sur la cinématique inverse (FABRIK)

Dans un instant bien défini, nous sélectionnons un frame dans le but de faire des mouvements supplémentaires (comme toucher une cible ou frapper quelque chose ou éviter un obstacle...etc). Nous utilisons la base des données des mouvements (utilisée dans le module 1) pour extraire les positions des jointures et les distances entre les jointures. Ces données représentant l'entrée de l'algorithme FABRIK.

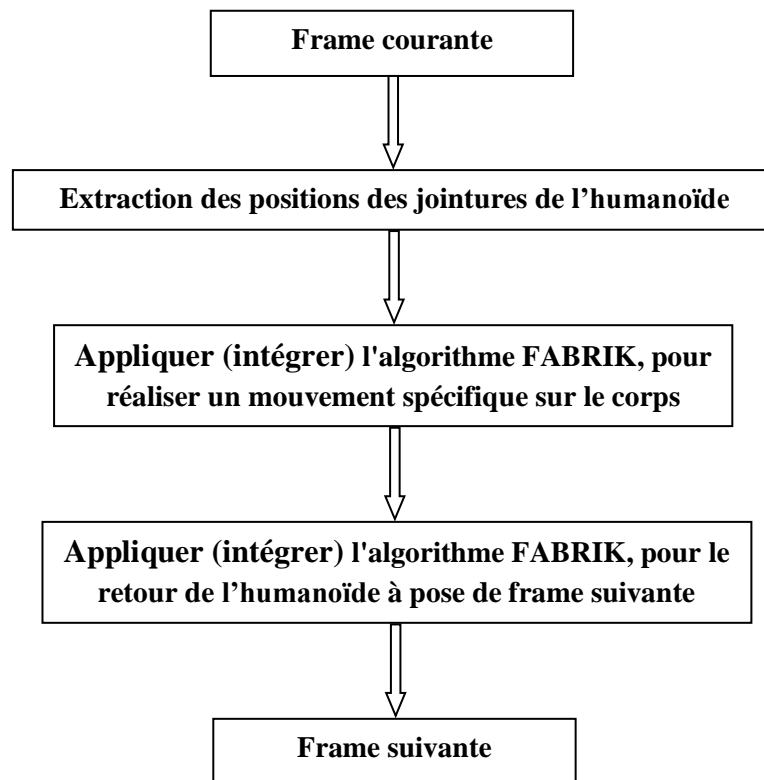


Figure 4.14: Appliquée FABRIK entre les frames, pour éditer nouveau mouvement.

b.1) Extraction des positions des jointures de l'humanoïde :

L'extraction des positions des jointures à cause de l'utilisation directe sur l'animation l'humanoïde par FABRIK.

Cette étape est résumée par l'algorithme suivant:

Algorithme 4.7 : Extraction les positions des jointures.

Entrées :

La position de la jointure racine **ROOT** ;

Sorties :

Tableau Position_Jointure ; /*position toutes les jointures fils de la jointure ROOT (x,y,z)*/

TQ n'est pas fin de la jointure **Faire**

/* calculer la position des jointures fils en basée sur jointure Parent */

TQ n'est pas fin de la jointure .fils **Faire**

Position_Jointure [jointure.fils.nom]=calculer_position_joint(jointure.Parent)

Fin TQ

Fin TQ

Fin

b.2) Appliquer FABRIK sur le corps de l'humanoïde :

L'humanoïde est divisé en six groupes, nous appliquons la cinématique inverse FABRIK à chaque groupe séparément,

Algorithme 4.8 : Ajustement complet du corps de l'humanoïde pour accéder à une cible.

Entrées :

La position de la jointure racine **ROOT** (**récupérer par frame courant**) ;

Pose courant d'humanoïde (**frame courant créer par module 1**)

Les effecteurs de fin (**récupérer par frame courant**).

Tableau Position_Jointure ;

Sorties :

Affichage nouveau position d'humanoïde;

Debut

Calculer la nouvelle **ROOT**,

Calculer La position du **Sterno-Clavicular** sur la base Les effecteurs de fin.

Ajustement du **tronc**, en utilisant la nouvelle position de **Sterno-Clavicular**.

Ajustement des **bras**.

Ajustement des **Jambes**.

Fin.

Fin.

Algorithme 4.9: détailler Algorithme 4.8.

Entrées :

La position de la jointure racine **ROOT**
 La position de cible **T**
 Pose courant d'humanoïde (frame courant créer par module 1)
 Les effecteurs de fin
 Tableau Position_Jointure

Sorties :

Nouveau position d'humanoïde

/* procédure pour calculer les longueur des membres */

D : entier **calcule_distance**(jointure.Nom)

Debut

Pour i=0 à jointure.fils **faire**

D+=jointure.nom → Distance (jointure.fils[i].nom)
 calcule_distance(jointure.fils[i].nom)

Fin pour

Return (D) ;

Fin

/* Ajustement complet du corps de l'humanoïde pour accéder à une cible par bras à gauche ou par jambe*/

Debut

Long_bras : longueur du bras

Long_jambe : longueur de la jambe.

Long_bras = calcule_distance (Rshoulder);

/* utilise **Rshoulder** ou **Lshoulder** pour calculé longueur du bras et
 utilise **Rhip** ou **Lhip** pour calculé longueur de la jambe */

Long_jambe = calcule_distance (**Rhip**);

/* distance entre cible T et la jointure Sterno-Clavicular */

Dist= Calcule_distance_cible (**T**, **Sterno-Clavicular**)

Si Dist <= Long_bras **alors**

Sterno-Clavicular et la Root garder leurs positions initiales

IK_FABRIK (Chaine3, T) ; /* appliqué FABRIK sur la chaine 3 pour atteindre la cible
 T (chaine2 le cas de bras à droite) */

Sinon

/* longueur du bras inferieur la distance entre cible T et la jointure Sterno-Clavicular */

Si jointure Sterno-Clavicular ne dépasser pas la limite d'orientation **alors**

/* Sterno-Clavicular contient 2 DDL **Figure 3.12** */

Chaine Nouv_Chaine = Rassemblage_les_Chaines(Chaine6, Chaine3)

/* connecter deux chaînes pour accéder la cible*/

IK_FABRIK (Nouv_Chaine, T).

Sinon /*la cible est très loin*/

Dist= Calcule_distance_cible (T, Rwrist)

/* distance entre cible T et la effecteurs de fin Rwrist */

Translation jointure Root avec le vecteur (posRoot_x+Dist, posRoot_y, posRoot_z) ;

Chaine Nouv_Chaine = Rassemblage_les_Chaines(Chaine6, Chaine3)

IK_FABRIK (Nouv_Chaine, T).

```

/* Nouv_effeteur_Fin fonction utilisé pour calculer la nouveau effecteur fin de la
jambe à gauche ou la jambe à droite, en fin utilisé la nouveau effecteur fin
comme cible par FABRIK pour reste l'animation réelle dans le cas de mise a
jour de ROOT */
IK_FABRIK (Chaine5, Nouv_effeteur_Fin(Lankle)).
IK_FABRIK (Chaine6, Nouv_effeteur_Fin(Rankle)).
/* Appliquer FABRIK sur chaque membre afin de trouver leur nouvelle configuration */
Fin si
Fin si
FIN
FIN

```

Dans le cas où il y a deux tâches (par exemple toucher cible par la main à droite et toucher d'autre cible par la main gauche), où les deux bras sont contraints influencés par la nouvelle configuration du tronc qui est calculée en tenant compte de la jointure Sterno-Clavicular qui va permettre de positionner les deux bras. Dans ces cas on utilise la priorité entre les tâches. Le classement des priorités des chaînes dérivées par les gestes naturels du corps humain.

c) Intégration de l'algorithme FABRIK, pour le retour de l'humanoïde à la pose de frame suivante.

Dans cette partie utilise les effecteurs de fin de la frame suivante comme cible.

Algorithme 4.10 : Ajustement complet du corps humanoïde pour retour à la position du frame suivant.

Entrées :

La position de la jointure racine **ROOT** (**récupérer par humanoïde**)
 Pose courant d'humanoïde (**créer par application du FABRIK**)
 Les effecteurs de fin
 Tableau Position_Jointure

Sorties :

Affichage nouveau position d'humanoïde
 /* Appliquer FABRIK sur chaque membre afin de retour a l'ancien configuration */
 Positionner la **ROOT** sur **ROOT** de la frame suivante.
 Calculer La position du **Sterno-Clavicular** par rapport les effecteurs de fin du frame suivante.
 Ajustement du **tronc**, en utilisant la nouvelle position de **Sterno-Clavicular**.
 Ajustement des **bras** par rapport Les effecteurs de fin du frame suivante.
 Ajustement des **Jambes** par rapport Les effecteurs de fin du frame suivante.

Fin.

4.4 Conclusion

La modification d'un mouvement est une tâche essentielle dans plusieurs domaines d'application. Nous avons présenté notre proposition qui sert à développer un système de modification du mouvement interactif en utilisant la cinématique inverse, ce dernier a comme entrée un seul mouvement. Il est constitué de deux étapes, la première sert à adapter le mouvement capturé à différents morphologies, alors que la deuxième sert à modifier interactivement le mouvement. L'idée principale de la méthode est de divisé le personnage virtuel en groupe relié entre eux par des jointures afin de garantir la corrélation entre les groupes du personnage ensuite nous appliquons la méthode FABRIK dans un ordre spécifique. Il faut aussi, choisir des jointures de connexion entre les groupes et appliqué la cinématique inverse (FABRIK) dans un ordre définie permet de respecter l'ordre des segments et donne des postures plus réaliste.

VI. Chapitre 5 : Mise en œuvre, résultats et bilan

5.1 Introduction

Ce chapitre a pour but l'expérimentation et la validation du modèle proposé dans le chapitre précédent, à travers la réalisation d'une animation. Pour ce faire, nous le divisons en trois parties. La première partie modélise un humanoïde virtuel en utilisant un logiciel graphique 3dsmax, la deuxième partie consisté à l'utilisation d'un mouvement capturé, existont dans le site mocap.cs.cmu.edu pour animer l'humanoïde virtuel, et troisième partie concerne l'intégration de la technique FABRIK pour modifier le mouvement de base (édition).

5.2 Environnement de développement

Nous avons développé notre application sous l'environnement suivant:

- Ordinateur portable : HP i5-4200 CPU 2.3 GHZ, RAM 4GO, Carte graphique 1GO, Disque Dure 460GO.
- Système d'exploitation : Microsoft Windows 8.1 édition intégrale 64bits.
- Environnement de développement : Microsoft Visual Studio 2010 Ultimate.
- Langage de programmation : C++.
- 3dsmax pour modélisation humanoïde.
- Moteur 3D OGRE pour animation.

5.2.1 Présentation de Microsoft VisualStudio 2010 Ultimate :

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.NET, des Services Web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C# et Visual C++ utilisent tous le même environnement de développement intégré (IDE), qui permet le partage d'outils et facilite la création de solutions à plusieurs langages. Par ailleurs, ces langages utilisent les fonctionnalités du .NET Framework, qui fournit un accès à des technologies clés simplifiant le développement d'applications Web ASP et de Services Web XML [66].

5.2.2 3DS MAX

3D Studio Max est un logiciel de modélisation et d'animation 3D, développé par la société Autodesk, il est très utilisé dans le monde de l'infographie, il a été utilisé dans la modélisation de l'humanoïde (squelette et revêtement) ainsi que l'environnement. La figure (5.1) nous montre modèle de l'humanoïde 3D [67].

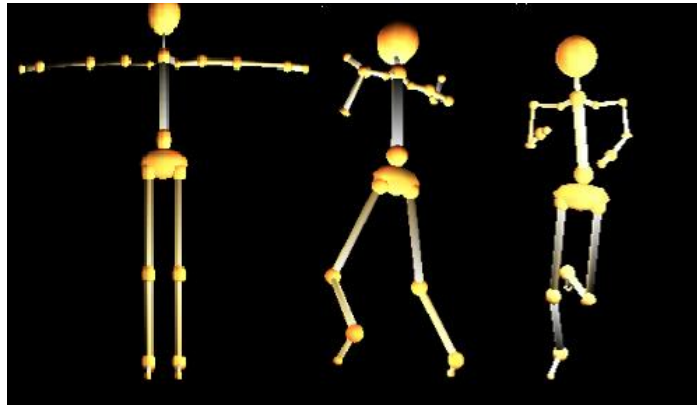


Figure 5.1 : Modèle de l'humanoïde 3D

5.2.3 Ogre 3D

OGRE signifie Object Oriented Graphics Rendering Engine, soit Moteur de Rendu Graphique Orienté-objet en français. OGRE est un moteur 3D open-source multi-plate forme (Linux, Win32, MacOSX, iPhone) orienté scène qui permet à partir d'objets à facettes de réaliser un environnement tridimensionnel qui sera perçu par un rendu bidimensionnel au travers d'une ou plusieurs caméra virtuelle. OGRE est entièrement écrit en C++ et de part sa conception, il permet aussi bien d'utiliser DirectX qu' OpenGL. Il intègre les dernières possibilités de ces deux bibliothèques et est en constante évolution. Par contre, OGRE est SEULEMENT un moteur 3D, il faut rajouter des modules pour gérer le son, la physique, le réseau... [68]

Historique

Le projet Ogre est né en 1999 de l'esprit d'un certain *Steve Streeting* [69], programmeur connu sous le pseudonyme de sinbad et qui a eu l'idée de construire un moteur 3D indépendant de la plateforme et de l'API utilisée (Direct3D ou OpenGL). Le projet est rapidement créé sur Source Forgeau début de l'année 2000 et

grossira progressivement sous la supervision de sinbad. Celui-ci se retire en Avril 2010 pour des raisons de santé mais reste un contributeur actif. La version 1.0.0 du moteur sort en février 2005 sous le nom de code Aza Thot. Les versions 1.x se succèdent au fil des ans et de façon régulière, toujours avec un nom de code désignant une créature mystique. En avril 2010 est sortie la version 1.7.1 dénommée Cthugha, qui en profite pour Changer de licence et passer en MIT en lieu et place de la LGPL et plusieurs versions ont suivi la version 1.7.1 comme les version 1.8.1 , 1.9 et les versions intégrées au Visual Studio.

Avantages de l'utilisation d'ogre :

- Multi plateforme : OGRE fonctionne aussi bien sous Windows, Linux ou MAC.
- Compatible OpenGL et DirectX.
- La plupart des plugins et outils sont tous développés pour VS.
- Très performant : il gère les effets graphiques les plus récents et affiche un Frame Rate très satisfaisant.
- Offre une architecture "professionnelle" : son design est tel qu'il peut parfaitement s'intégrer à des applications graphiques professionnelles. Le travail en équipe ne pose pas de problème grâce à la POO et sa modularité offre de grandes possibilités de personnalisations.
- Très bien documenté : OGRE offre une API de très bonne qualité, un manuel complet et des tutoriaux tous très bien fait. Le forum est aussi très actif et peu beaucoup aider.

5.2.3.1 L'entité

Une entité, c'est la représentation dans la scène d'un modèle 3D, aussi appelé mesh (mesh signifie maillage en anglais). Le mesh est l'ensemble des polygones élémentaires (des quadrilatères et des triangles, le plus souvent) que l'on crée dans un logiciel de modélisation, comme 3DS Max ou Blender ou Maya, et qui constituent votre modèle 3D complet. Tous ces polygones sont reliés entre eux par leurs sommets. Plus il y a de sommets, et donc de polygones, plus le mesh est précis. Généralement, le mesh possède aussi une ou plusieurs textures (une image plaquée sur les polygones) qui lui donnent un aspect plus réaliste qu'une couleur unie lors du rendu. Enfin, pour certains modèles comme les personnages, le mesh possède un squelette qui permet de créer des animations pour son mouvement. Les mesh sont gérés par le gestionnaire MeshManager. Ils

sont généralement chargés à partir du format de fichier spécifique à OGRE: le format ".mesh". Ces fichiers sont créés grâce aux outils d'exportation (section 2.3.2) disponibles dans la plupart des logiciels 3D, comme ils peuvent être créés manuellement en appelant la méthode **MeshManager::createManual**.

5.2.3.2 Le SceneManager

Afin de pouvoir gérer l'ensemble des objets de notre scène tout au long du déroulement du programme, OGRE introduit la classe de gestion de la scène "SceneManager".

Concrètement, le SceneManager s'occupe de garder une trace de tous les modèles, lumières, caméra et autres objets que peut contenir la scène. C'est à lui que revient la tâche de créer tous ces objets et de nous permettre ensuite d'y accéder. Par conséquent, l'insertion d'objets dans la scène passe toujours par lui (ou par l'un des éléments déjà insérés) et par les méthodes qu'il propose pour cela. Il en existe diverses variantes en fonction de la scène que l'on veut réaliser ; selon que celle-ci sera en intérieur ou en extérieur, par exemple, on pourra utiliser un SceneManager différent.

5.2.3.3 Ogre Exporter

Les exportateurs sont des plugins de modèleur 3D qui écrivent le maillage et le squelette d'animation et le texture dans un format de fichier compréhensible par OGRE. Ces fichiers sont respectivement au format **.mesh**, **.skeleton** et **.material**.

Nous allons utiliser OgreMax Scene Exporter. Il est le plugin du modèleur 3DSMax utilisé pour l'exportation des scènes de format compréhensible par 3DSMax vers des scènes de format compréhensible par OGRE. Nous avons utilisé la version 2.6.4 du 20 octobre 2014.

5.2.4 Représentation de l'interface développée

Nous avons proposé dans la partie précédente deux modèles de l'animation d'humanoïde, un modèle avec capture de mouvement et l'autre avec capture de mouvement et la technique FABRIK.

Dans la première version de notre application, nous avons animé notre humanoïde sans interaction avec l'environnement (figure 5.2).

Dans la deuxième version, nous avons animé notre humanoïde sur l'environnement et avec interaction des objets dans l'environnement (figure 5.3).

Modélisation d'un environnement virtuel chambre avec des murs et une table.

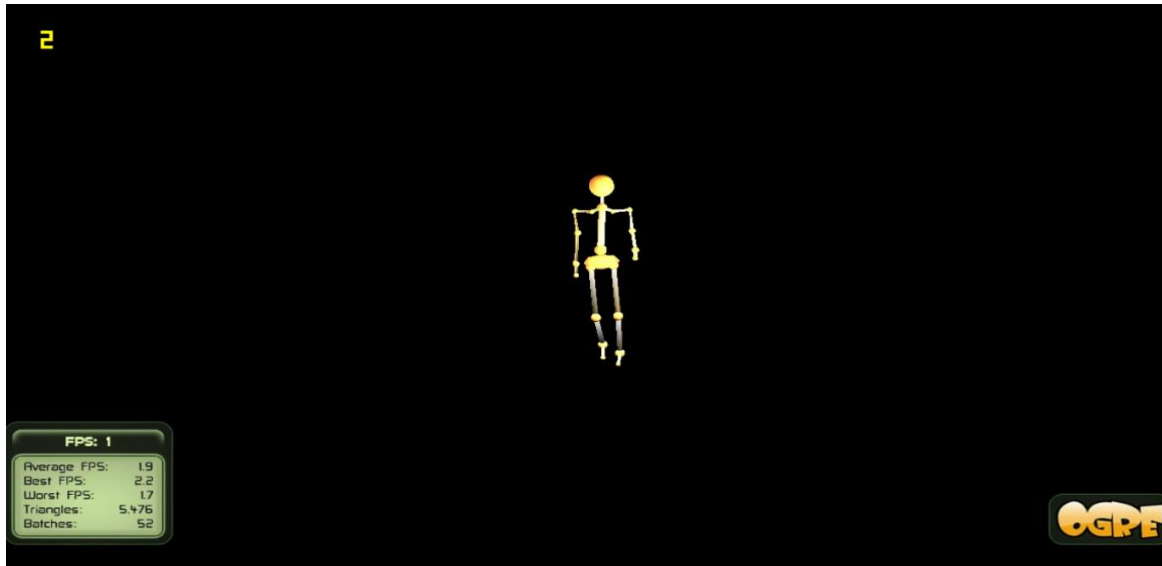


Figure 5.2 : L'interface de première version d'application.

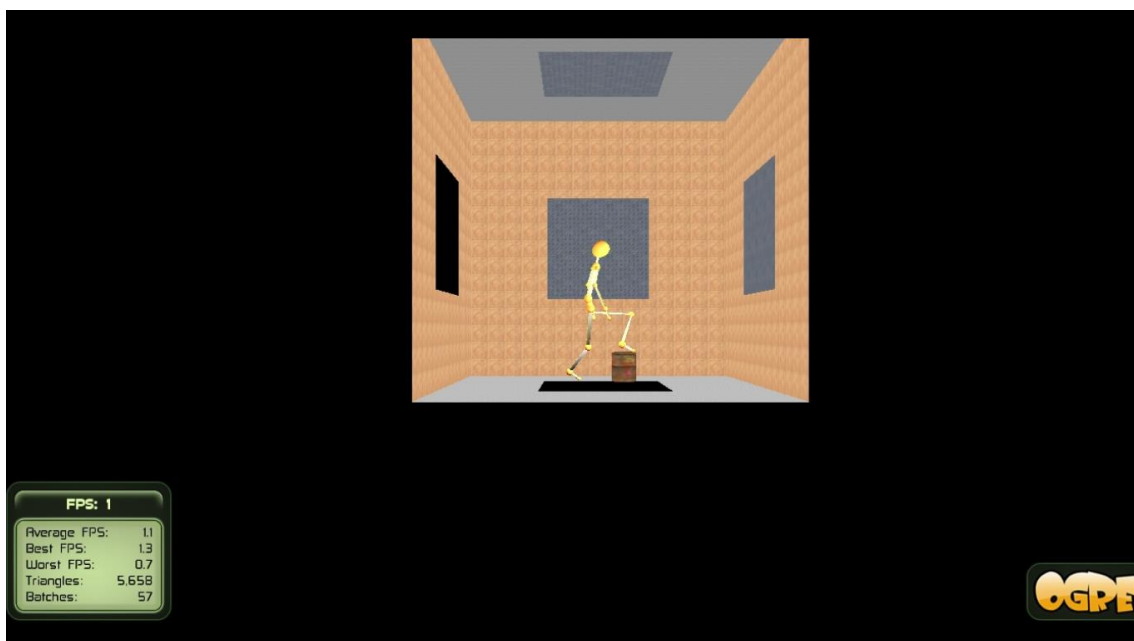


Figure 5.3 : L'interface de deuxième version d'application.

Dans les deux versions, en représente les informations nécessaires pour l'état de la simulation d'animation (Taux moyen d'affichage, Nombre de triangles affichés) figure 5.4.



Figure 5.4: Information sur l'état de la simulation.

5.3 Utilisation d'un mouvement capturé pour animer l'humanoïde :

Dans ce partie nous avons utilisé, une capture de mouvement seule pour l'animation de l'humanoïde, avec les fichiers captures des mouvements (*.bvh) dans le site « mocap.cs.cmu.edu et d'autres sites », l'objectif est d'adapter un mouvement déjà capturé d'un personnage à un autre, ayant différentes topologies et différentes géométries.

5.3.1 Mise en correspondance entre les jointures de fichier du mouvement et le personnage virtuel

Cette étape est essentielle dans tous les systèmes d'adaptation du mouvement, dans notre proposition pour cette étape, nous avons choisi d'appliquer un fichier de mappage squelette qui se base sur les noms des jointures figure 5.5.

```
// skeleton mapping file
BVH1:Hips Plasticman1:Hips
BVH1:RightUpLeg Plasticman1:RightUpLeg
BVH1:RightLeg Plasticman1:RightLeg
BVH1:RightFoot Plasticman1:RightFoot
BVH1:RightToeBase Plasticman1:RightToeBase
BVH1:RightToeEnd Plasticman1:RFoot_End
BVH1:LeftUpLeg Plasticman1:LeftUpLeg
BVH1:LeftLeg Plasticman1:LeftLeg
BVH1:LeftFoot Plasticman1:LeftFoot
BVH1:LeftToeBase Plasticman1:LeftToeBase
BVH1:LeftToeEnd Plasticman1:LFoot_End
BVH1:Spine Plasticman1:Spine1
BVH1:Head Plasticman1:Head
BVH1:HeadEnd Plasticman1:Head_End
BVH1:RightShoulder Plasticman1:RightShoulder
BVH1:RightArm Plasticman1:RightArm
BVH1:RightForeArm Plasticman1:RightForeArm
BVH1:RightHand Plasticman1:RightHand
BVH1:LeftShoulder Plasticman1:LeftShoulder
BVH1:LeftArm Plasticman1:LeftArm
BVH1:LeftForeArm Plasticman1:LeftForeArm
BVH1:LeftHand Plasticman1:LeftHand
```

Figure 5.5: Fichier de mappage squelette (BVH1 : nom du fichier du capture de mouvement, Plasticman1 : nom d'humanoïde).

5.3.2 Résultats

Dans cette partie nous allons présenter les résultats obtenus après l'exécution de notre système d'animation de personnage virtuel par les fichiers de capture de mouvements, qui sont implémentés et testés sur plusieurs types des mouvements.

5.3.2.1 Adaptation du mouvement de la marche

Dans cette expérimentation, le mouvement d'origine est un mouvement capturé avec 190 frames. Nous avons appliqué le mouvement de la marche sur notre personnage virtuel figure 5.6, nous allons appliquer la méthode d'adaptation sur la partie inférieure qui consiste de pelvis, cuisse, jambe et la partie supérieure qui consiste de tête, tronc, bras, avant-bras.

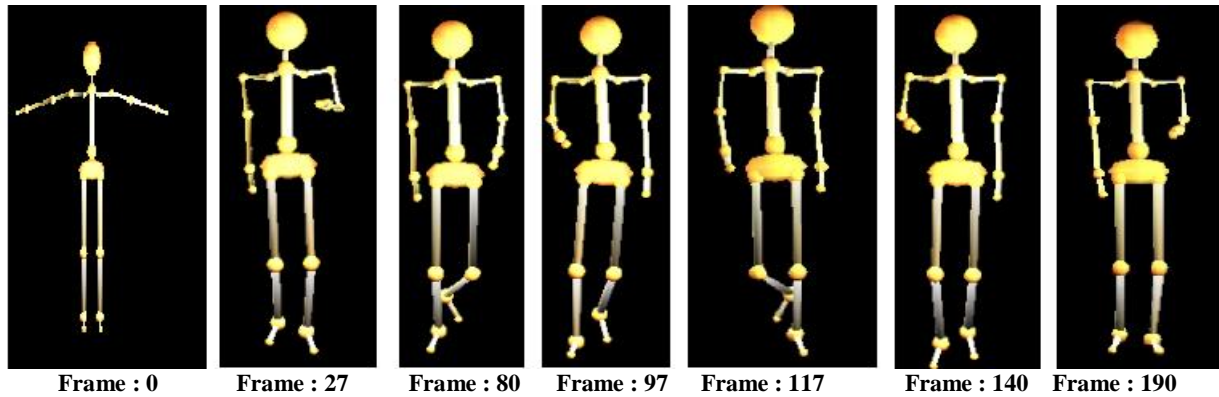


Figure 5.6 : Adaptation du mouvement de la marche

5.3.2.2 Adaptation du mouvement de la course

Le mouvement d'origine est un mouvement capturé avec 142 frames. Nous avons appliqué le mouvement de la course sur notre personnage virtuel figure 5.7

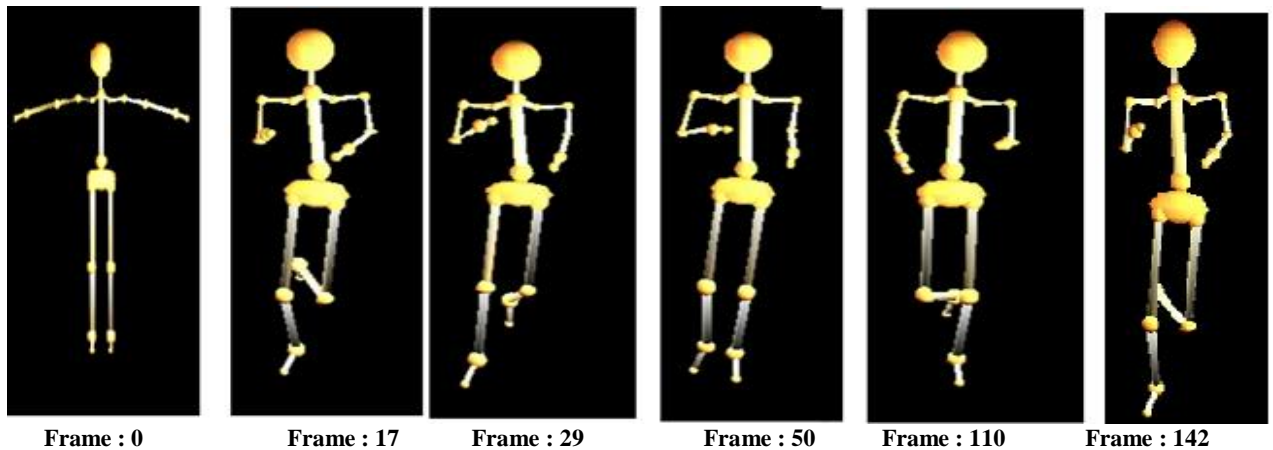


Figure 5.7 : Adaptation du mouvement de la course

5.3.2.3 Adaptation du mouvement de la boxe

Le mouvement d'origine est un mouvement capturé avec 3069 frames. Nous avons appliqué le mouvement de la boxe sur notre personnage virtuel figure 5.8, et nous avons appliqué le même mouvement sur deux différents personnages virtuels figure 5.9.

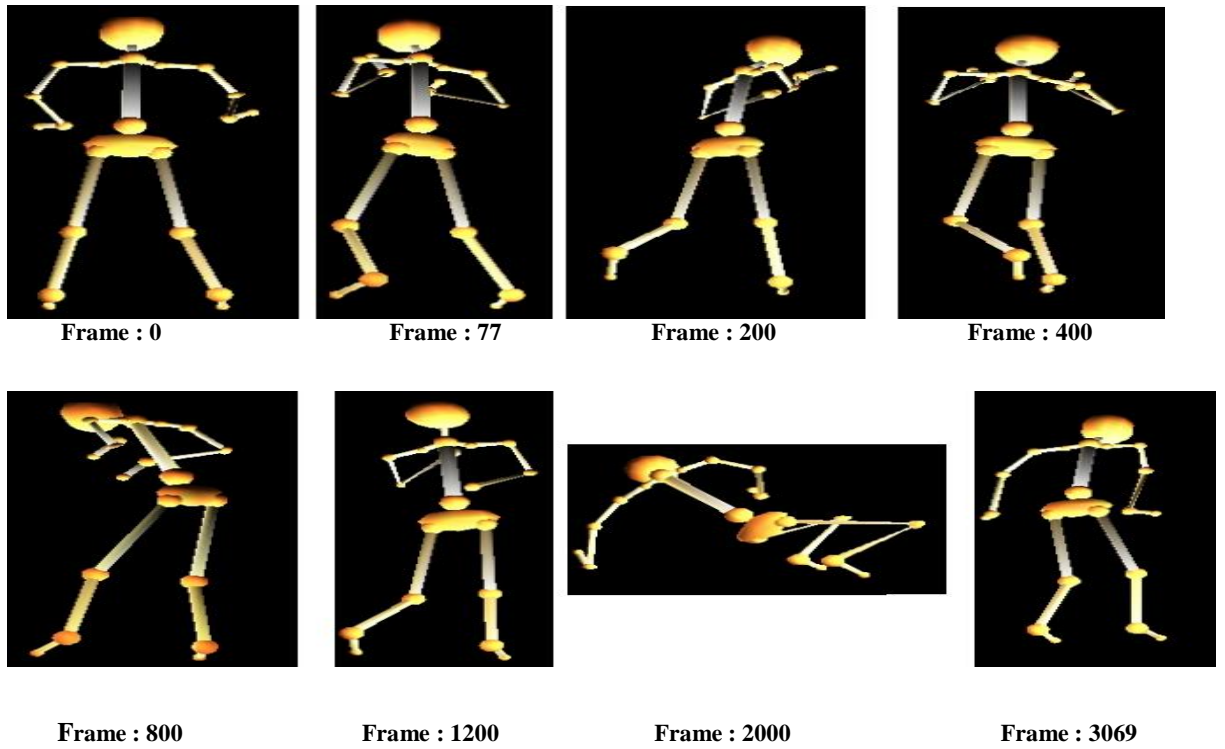


Figure 5.8 : appliqué le mouvement de la boxe sur notre personnage virtuel.

5.3.2.4 Adaptation du mouvement de la boxe sur deux différents personnages virtuels

Le personnage de final est réduit, la comparaison entre les dimensions est donnée dans le tableau suivant :

Membre	Bras	Avant-bras	Tronc	Pelvis	Cuisse	Jambe	Pied
Origine	32	38	42	7	48	51	22
Final	18	22	25	7	32	38	10

Table 5.1 : Comparaison entre les dimensions du personnage virtuel.

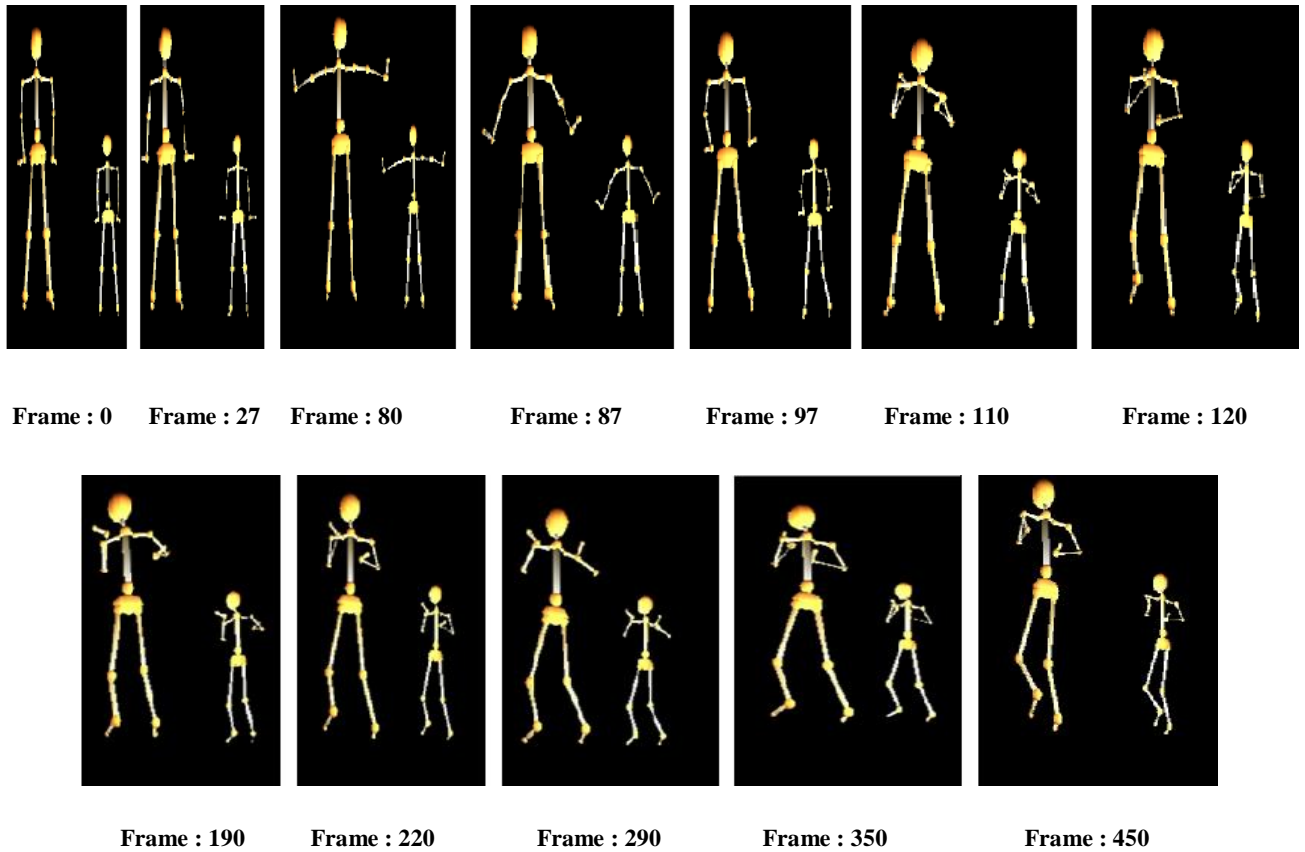


Figure 5.9 : Application du même mouvement de la boxe sur les deux personnes.

Une comparaison entre le mouvement d'un personnage d'origine et un personnage plus grand est donnée dans la table 5.1, nous avons activé des contraintes sur la main et le pied du personnage de destination depuis le frame 350. Depuis la table, nous constatons que les angles d'une posture de personnage d'origine et posture de personnage final sont presque les mêmes, cela est due à l'utilisation d'une méthode d'animation par capture des mouvements qui prend en charge le respect des limites articulaires.

	Angle Posture de pers d'origine	Angle posture de pers final
Angle1_Arm_Left	-9.454	-9.455
Angle2_Arm_Left	4.319	4.316
Angle3_Arm_Left	-61.905	-61.900
Angle1_Foot_Left	0.284	0.288
Angle1_Foot_Left	62.513	62.512
Angle1_Foot_Left	4.128	4.128

Table 5.1 : Comparaison entre les angles du personnage virtuel.

5.4 Application (intégration) de FABRIK pour modifier le mouvement de base (édition).

Dans ce partie, on utilise la méthode FABRIK sur l'animation réalisé par capture mouvement pour changer mouvement de base (édition du mouvement).

5.4.1 Résultats

Dans cette partie nous allons présenter les résultats obtenus après l'exécution de notre système d'animation de personnage virtuel par les fichiers de capture de mouvements et intégré FABRIK pour atteint un objet. Ceux ci sont implémentés et testés sur plusieurs types des mouvements avec différentes contraintes.

5.4.1.1 Adaptation du mouvement de la marche pour atteint un objet

Dans cette section, nous allons présenter quelques résultats de notre méthode. Les figures 5.11, 5.12 représente respectivement l'atteinte d'un objet avec le bras seulement, et avec les deux bras, en spécifiant la posture initiale du personnage virtuel ainsi que la cible visé(en rouge). Nous avons récupéré un mouvement de marche contient 3199 frames avec un intervalle de 0.01 de temps entre les frames successives.

- **Toucher un objet en utilisant le bras à gauche**

Dans le premier test 5.11, la cible se trouve à proximité du personnage, dans ce cas le tronc garde sa configuration initiale et le bras change de configuration. La posture résultat est naturelle parce que lorsque la cible est proche nous utilisons seulement le bras pour l'atteindre. En utilisant cette stratégie, nous minimisons le temps de calcul et l'énergie pour atteindre un objet.

Dans ce partie appliqué, nous appliquons FABRIK sur la chaine 3 (le bras à gauche), nous avons choisi frame 300 comme début d'intégration de FABRIK, (frame 350) le personnage virtuel atteint un objet.

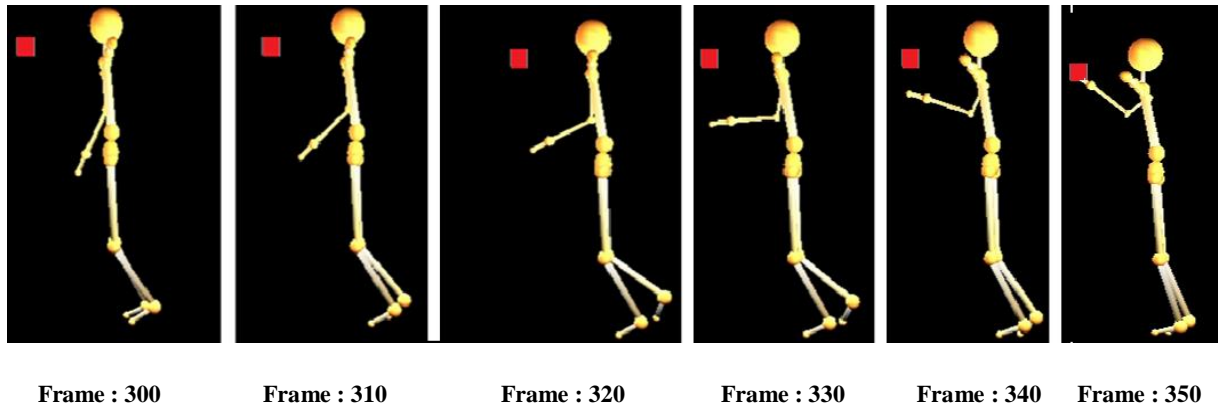


Figure 5.10: FABRIK utilise une seule chaîne pour atteindre un objet

- **Toucher un objet en utilisant les deux bras**

Dans cette partie nous appliquons FABRIK sur la chaîne 2 (le bras à droite) et la chaîne 3 (le bras à gauche), nous avons choisi frame 300 comme début d'intégration de FABRIK, (frame 360) le personnage virtuel atteint un objet (la figure 5.12).

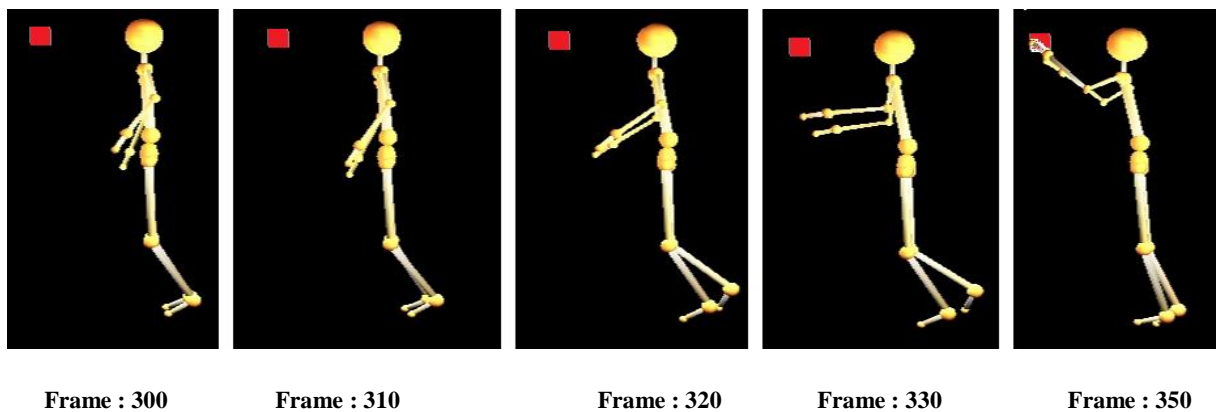


Figure 5.11: FABRIK utilise deux chaînes pour atteindre un objet

Dans la deuxième version, nous avons modélisé d'un environnement virtuel chambre avec des murs et un terrain. Nous avons animé notre humanoïde sur l'environnement (contact entre les pieds et le sol tout au long du mouvement), interaction des objets dans l'environnement (figure 5.13).

Dans cette partie nous appliquons FABRIK sur la chaîne 2 (le bras à droite) et la chaîne 3 (le bras à gauche), l'ajustement complet du corps de l'humanoïde pour garder les pieds sur terre, est appliqué l'algorithme 4.8.

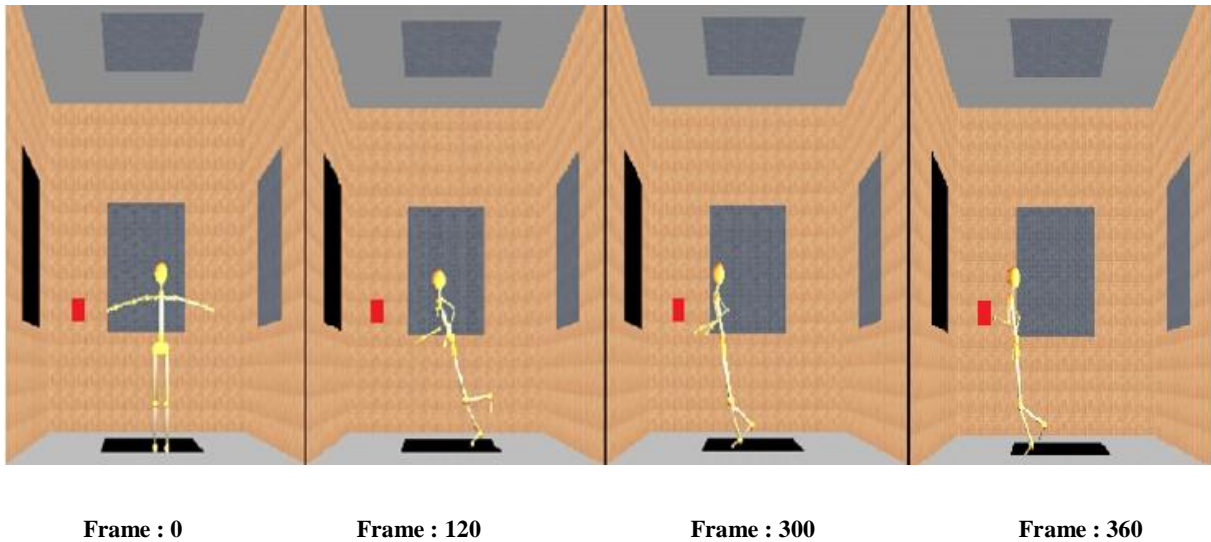


Figure 5.12: FABRIK utilise deux chaînes pour atteindre un objet, gardez les pieds sur terre.

- **Toucher un objet et éviter la collision en utilisant le pied à droite**

Dans cette expérimentation, le mouvement d'origine est un mouvement capturé avec 600 frames. Comme la partie inférieure du corps est plus importante dans le mouvement de la marche, nous allons appliquer la méthode d'adaptation seulement sur la partie inférieure qui consiste de pelvis, cuisse, jambe et pied pour atteindre un objet seulement avec le pied.

Dans ce partie nous avons appliqué FABRIK sur la chaîne 4 (la jambe à droite), choisi frame 320 le début d'intégrer FABRIK, (frame 350) le personnage virtuel atteint un objet (la figure 5.14).

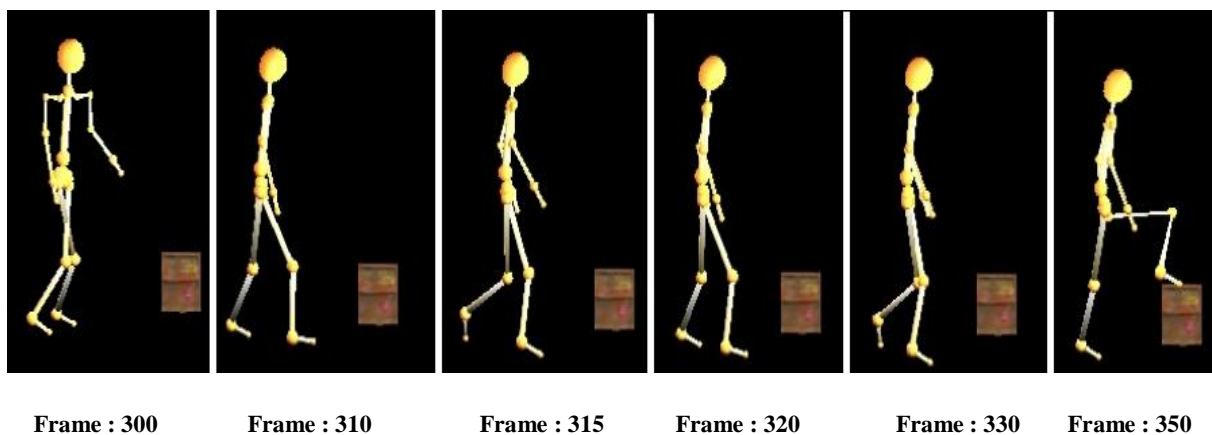


Figure 5.13: Atteindre un objet, en utilisant uniquement le pied.

Dans la deuxième version, nous avons modélisé un environnement virtuel chambre avec des murs et un terrain, et nous avons animé notre humanoïde sur l'environnement (contact entre les pieds et le sol tout au long du mouvement), la cible exacte visé(en rouge) figure 5.14. Éviter la collision avec la table maron figure 5.15.

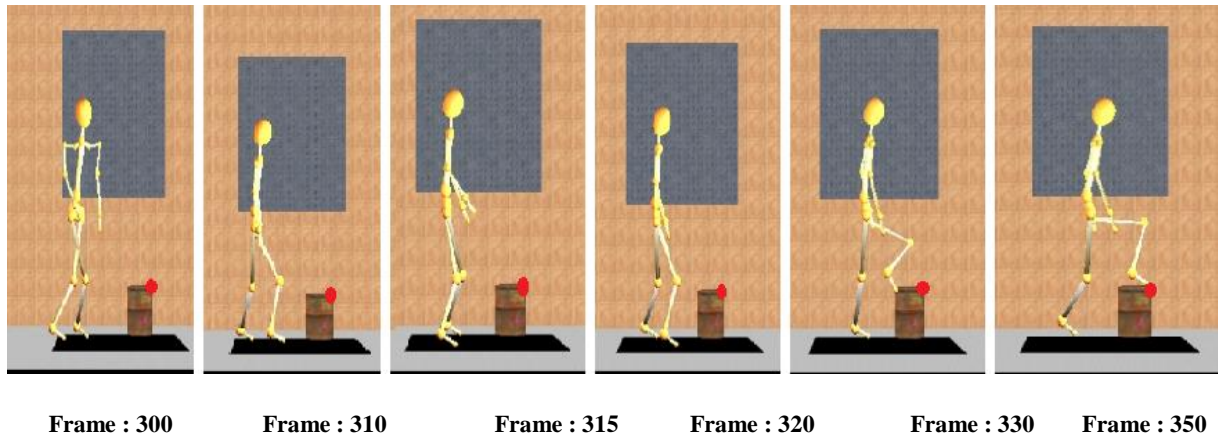


Figure 5.14: Les pieds de personnage sont sur le sol et l'objet est atteint.

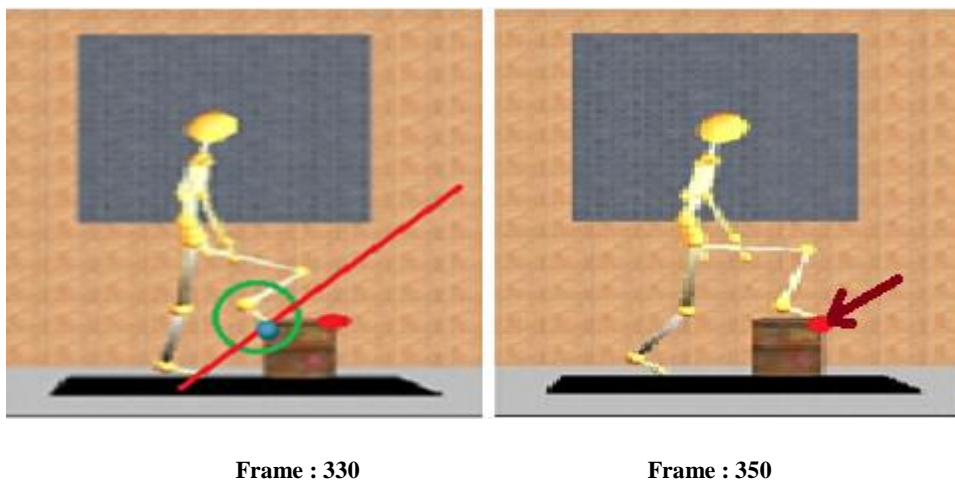


Figure 5.15: Eviter la collision entre le pied et la table.

Dans la figure 5.15 nous utilisons le point bleu comme première cible pour éviter la collision avec la table. Ensuite, nous utilisons le point rouge deuxième cible pour meilleur position du pied sur la table.

- **Evaluation**

Afin d'élaborer notre méthode, nous l'avons soumis à un jeu de tests en variant la complexité des humanoïdes à base de triangles 5000 à 12000 triangles pour mesurer la fréquence d'affichage des images par seconde: FPS.

	MOCAP	MOCAP	MOCAP	MOCAP	MOCAP	MOCAP	FABRIK	FABRIK	FABRIK	FABRIK	FABRIK	FABRIK
Triangles	5480	5546	5604	5658	10100	12000	5480	5546	5604	5658	10100	12000
FPS 1	25	22	19	16	12	9	45	27	24	16	13	11
FPS 2	26	20	19	14	12	7	45	25	22	16	13	12
FPS 3	21	19	16	12	11	7	29	23	22	16	12	10
FPS 4	23	18	14	11	10	6	21	26	20	16	11	9
FPS 5	24	19	14	11	4	6	20	26	15	16	10	9
FPS Moyenne	23,8	19,6	16,4	12,8	9,8	7	32	25,4	20,6	16	11,8	10,2

Table 5.2 : Comparaison entre une posture capturé et une autre construite par la méthode proposée (mélange entre capture mouvement et FABRIK)

Pour comparer les méthodes, nous avons choisi le critère FPS (Frame Per Second) pour évaluer les performances. Le tableau 5.2 représente une comparaison entre une posture capturé et une autre construite par la méthode proposée (mélange entre capture mouvement et FABRIK). Les résultats obtenus (figure 5.16) montre une nette amélioration en FPS moyenne de la méthode FABRIK par rapport à la méthode classique MOCAP, pour le même nombre de triangles.

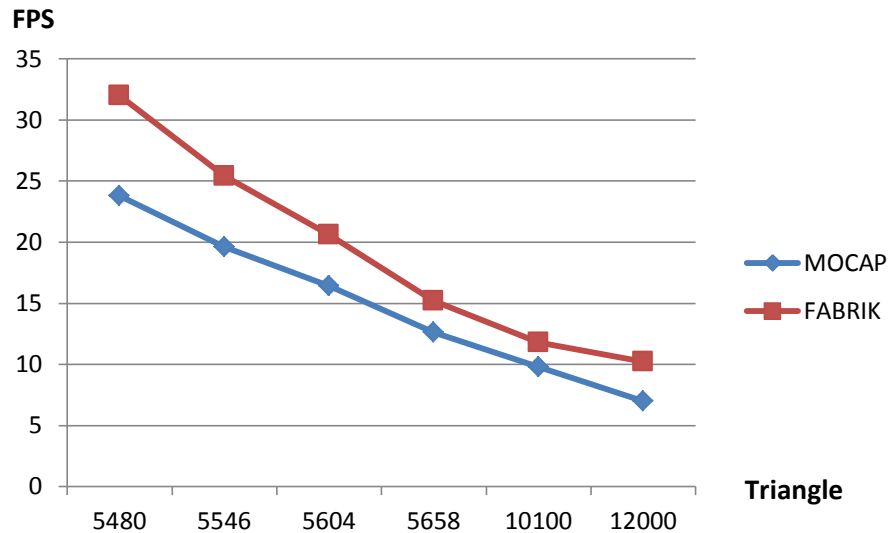


Figure 5.16: Graphe des valeurs des FPS par rapport au nombre de triangles entre posture capturé et une autre construite par la méthode proposée.

	MOCAP	MOCAP	MOCAP	MOCAP	MOCAP	MOCAP	FABRIK	FABRIK	FABRIK	FABRIK	FABRIK	FABRIK
Triangles	5480	5546	5604	5658	10100	12000	5480	5546	5604	5658	10100	12000
Temps de calcul (Secondes)	60	72,6	75	90	180	204	42	60	66	79,8	138	180

Table 5.2 : Temps de calcul mesuré par rapport au nombre de triangles.

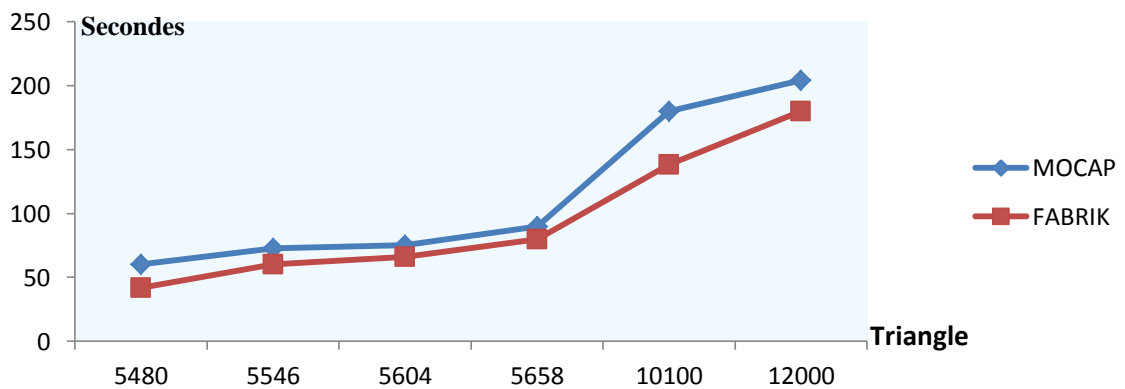


Figure 5.17: Graphe du temps de calcul mesuré par rapport au nombre de triangles entre posture capturé et une autre construite par la méthode proposée.

Le graphe de la figure 5.17 illustre les variations du temps de calcul par rapport aux nombres de triangles dans l'animation, on observe que le temps de calcul est largement avantageux pour la méthode à base FABRIK.

5.5 Conclusion

L'édition de mouvement est une étape essentielle afin d'adapter un mouvement à une autre morphologie et/ou à un autre environnement. Plusieurs méthodes ont été proposées dans la littérature, parmi elles les méthodes reposant sur la cinématique inverse (IK). La plupart des méthodes de IK souffrent : de temps de calcul, de la complexité et du manque de réalisme.

Dans ce chapitre nous avons parlé de l'environnement de développement et des différents logiciels de programmation utilisés pour assurer l'animation d'humanoïde virtuel de manière plus naturelle et proche de la vie réelle des humains, puis nous avons défini les entrées de notre système « fichier des captures mouvement, modèle humanoïde virtuel », ensuite nous avons présenté les résultats d'animations par capture des mouvements, enfin les résultats d'animations par méthode FABRIK et captures mouvement. L'intégration de la méthode FABRIK pour l'animation et l'édition de mouvement a montré son efficacité dans l'optimisation du temps de calcul, ce qui nous a permis d'obtenir la génération des scènes en temps réel.

Conclusion générale

Nous nous sommes intéressé dans ce mémoire de magister à la simulation d'animation d'humain virtuel, en exploitant la notion de capture de mouvement pour génération de mouvement et cinématique inverse comme technique d'édition de mouvement. Nous avons concentré notre effort sur la méthode FABRIK ainsi que sur les captures des mouvements.

Actuellement, les chercheurs se focalisent dans leurs travaux dans le domaine de l'animation des humanoïdes de synthèse sur le problème de temps de calcul, donc le besoin d'algorithmes moins coûteux en temps de calcul. L'intégration de la méthode FABRIK avec capture du mouvement, offre cette propriété et préserve les caractéristiques importantes d'un mouvement initial.

Dans ce cadre, nous avons développé plusieurs points :

1. Création de la scène : un outil de création de scènes virtuelles a été proposé. Il permet d'ajouter, modifier, repositionner, réorienter ou supprimer des objets dans l'environnement et importer un fichier de mouvement de type BVH et lui ajouter les informations de la scène.
2. Edition du mouvement : C'est l'étape la plus cruciale dans notre proposition. Le but de cette étape est d'importer des mouvements, puis de les modifier afin d'obtenir d'autres plus complexes ou qui répondent aux exigences de l'utilisateur. Pour obtenir ces mouvements, nous avons proposé :
 - (a) Cinématique inverse : Après une étude détaillée sur les méthodes existantes dans le domaine d'édition de mouvement, nous avons pu conclure que les méthodes à base de la cinématique inverse (FABRIK) répondent aux spécificités de notre cas d'étude. Cependant, ces méthodes utilisent une base de mouvement afin de modifier un mouvement donné.
 - (b) Diviser le personnage virtuel en groupes reliés entre eux par des jointures afin de garantir la corrélation entre les groupes du personnage, ensuite nous appliquons la méthode FABRIK dans un ordre spécifique, la division du personnage en groupe facilite le calcul de la solution.

- (c) Adaptation du mouvement à une autre morphologie : nous avons choisi d'appliquer un fichier de mappage squelette qui se base sur les noms des jointures.

Perspectives

Le travail que nous avons présenté dans ce mémoire, est la première étape pour la mise en place d'un système complet de capture, édition et modification de mouvement, pour cela nous pensons que d'autres sujets de recherche peuvent être proposés afin de l'améliorer et de l'étendre. Les perspectives que nous pouvons prévoir consiste à :

- obtenir une animation complète du corps, pourquoi pas exploiter les fichiers qui contiennent l'animation de visage (expression facial).
- Intégrer l'animation de personnage virtuel par la capture de mouvements et cinématique inverse dans des applications comme la simulation et la gestion de foule des piétons.
- Améliorer l'apparence d'animation et ajoutant des effets à la scène, comme l'ombre, l'effet de transparence et calculer l'éclairage.
- Intégrer le calcul par les GPUs, ce qui permettra d'avoir une plus grande fluidité dans l'animation des scènes.

Bibliographie

- [1] J.E. Chadwick, D.R. Haumann & R.E. Parent, "Layered Construction for Deformable Animated Characters". In ACM Computer Graphics (Proc. of SIGGRAPH'89), pages 179-188, 1989.
- [2] Jean Dif, "Modélisation du corps humain en 3D". Vuibert, Paris, page 355, 1998.
- [3] Tolga Capin, I. S. Pandzi c, N. Magnenat Thalmann, D. Thalmann, "Realistic Avatars and Autonomous Virtual Humans in VLNET Networked Virtual Environments". Computer Graphics Laboratory, 1999.
- [4] M.Meredith, S.Maddock, "Motion Capture File Formats Explained". Department of Computer Science, University of Sheffield.
- [5] Catherine POTEL, Philippe GATIGNOL, "Laissons mécaniques modélisation". Université du Maine le Mans.
- [6] H. Gray, "Editeur Anatomy of the human body". Lea & Febiger, 1918.
- [7] W. Maurel, D. Thalmann, "Human Upper Limb Modeling including Scapulo-Thoracic Constraint and Joint Sinus Cones". Computers and Graphics, vol. 24, no. 2, pages 203-218, 2000.
- [8] A. Maciel, L. Porcher Nedel et C. M. Dal Sasso Freitas, "Anatomy-Based Joint Models for Virtual Human Skeletons". In CA02 : Proceedings of the Computer Animation, page 220. IEEE Computer Society, 2002.
- [9] J. Wilhel, "Animals with Anatomy". IEEE Computer Graphics and Applications, vol. 17, no. 3, pages 22-30, May 1997.
- [10] A. Aubel et D. Thalmann, "Interactive Modeling of the Human Musculature". Proceedings of Computer Animation, pages 167-173 , 2001.
- [11] Rick Parent, " Animatique Algorithmes et techniques". Vuibert, Paris,2003.
- [12] N. Burtnyk, M. Wein, "Computer Generated Key Frame Animation", Journal of the Society of Motion Picture and Television Engineers, pages 149-153, 1971.
- [13] Ken Perlin, Athomas Goldberg, "A system for scripting interactive actors in virtual worlds". SIGGRAPH Proceedings of the 23 rd annual conference on Computer graphics and interactive techniques, pages 205-216, 1996.
- [14] Ken Perlin, "Real Time Responsive Animation with Personality". IEEE Transactions on Visualization and Computer Graphics , Page 5-15, 1995.

- [15] Jui-Fa Chen, Wei-Chuan Lin, Kun-Hsiao Tsai , Shih-Yao, Dai "Analysis and Evaluation of Human Movement based on Laban Movement Analysis". Tamkang Journal of Science and Engineering, Vol. 14, No. 3, pages 255 -264 ,2011.
- [16] B. le Callennec, "interactive techniques For motion deformation of articulated figures Using prioritized constraints", thèse de doctorat, faculté d'informatique et communications, école polytechnique fédérale de Lausanne, page 19,2006.
- [17] N. Courty (VALORIA) G. Dumont (IRISA) et al ,AS Huma in Virtuel, " thème 2 : Mouvement contrôle de mouvement ".
- [18] H. Rijkema ,M. Girard, "Computer animation of knowledge-based human grasping". In Proceedings of ACM SIGGRAPH, pages 339-348, Juillet 1991.
- [19] A. Maciel, L. Porcher Nedel, C. M. Dal Sasso Freitas, "Anatomy-Based Joint Models for Virtual Human Skeletons". In CA'02 : Proceedings of the Computer Animation, page 220. IEEE Computer Society, 2002.
- [20] D. Furukawa, " Human Spine Posture Estimation from Video Images Based on connected Vertebra Spheres Model". In Proceedings of the 14th Conference on Computer Animation'01, pages 174-181, 2001.
- [21] Corlouer. F, Bellamine. E, Gabet. S, &Dumartin. T "Analyse de mouvement humain 3D à base d'extraction de squelette géométrique". Projet de Programmation, Mémoire Final, Université Bordeaux, vol 1. 2008.
- [22] Marko Jug, Janez Pers, Branko Dezman, and Stanislav Kovacic. "Trajectory based assessment of coordinated human activity". In3rd International Conference on Computer Vision Systems, pages 534-543, 2003.
- [23] Tournier. M " Paramétrisation de données de capture de mouvement pour la compression d'animations 3D". Master thesis 2, Institut National Polytechnique Grenoble (INPG) ,2007.
- [24] Salmeron Quiroz, Bernardino Benito "Fusion de données multicateurs pour la capture de mouvement", Doctoral dissertation, Université Joseph-Fourier-Grenoble I, 2007.
- [25] Schönauer.C, "skeletal structure generation for optical motion capture". Institute for Software Technologies and Interactive Systems, Vienna University of Technology.2007.

- [26] Dennis Holmboe, "The Motion Capture Pipeline". Thèse de doctorat. University of Gävle. 2008.
- [27] G. Johansson, "Visual perception of biological motion and a model for its analysis". *Perception and Psychophysics*, pages 201–211, 1973.
- [28] M. Meredith, S. Maddock, "Motion Capture File Formats Explained", Department of Computer Science, University of Sheffield.
- [29] Ashraf.G et Wong.K, "Constrained Framespace Interpolation", *Computer Animation*, pages 61-72, 2001.
- [30] Kovar.L, Gleicher.M, and Pighin, F. "Motion Graphs", *ACM Transactions on Graphics*, pages 473-482, 2002.
- [31] Andrew Witkin and Michael Kass, "Spacetime constraints". In John Dill, editor, *Computer Graphics (SIGGRAPH Proceedings)*, volume 22, pages 159–168, 1988.
- [32] Michael F. Cohen, "Interactive spacetime control for animation". In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 293–302, 1992.
- [33] M. Gleicher, "Retargetting motion to new characters". In *SIGGRAPH 1998*.
- [34] Fang (A.C) et Pollard (N.S) "Efficient Synthesis of Physically Valid Human Motion". *Proceedings of SIGGRAPH*, vol. 22, pages 417-426, 2003.
- [35] Zordan.V.B, Majkowska.A, Chiu.B et Fast.M, "Dynamic Response for Motion Capture Animation". *Proceedings of SIGGRAPH*, vol. 24, no. 3, pages 697-701, 2005.
- [36] Ménardais.S, "Fusion et adaptation temps réel de mouvements acquis pour l'animation d'humanoïdes synthétiques". PhD thèses, IRISA Rennes, 2003.
- [37] Multon.F, Kulpa.R et Bideau.B, "A Global Framework for Animating Humans in Virtual Reality Applications". *Presence*, vol. 17, no. 1, pages 17-28, 2008.
- [38] Monzani.J.S, Baerlocher.P, Boulic.R et Thalmann .D, "Using an Intermediate Skeleton and Inverse Kinematics for Motion Retargeting". *Proceedings of Eurographics*, 2000.
- [39] Choi (K.J) et Ko (H.S), "On-line motion retargeting". *The Journal Of Visualisation and Computer Animation*, vol. 11, no. 5, pages 223-235, 2000.

- [40] Pronost.N, Dumont.G, Berillon.G & Nicolas.G, "Morphological and stance interpolations in database for simulating bipedalism of virtual humans".*Visual Computer*, vol. 22, pages 4-13, 2006.
- [41] Yamane (K), Kuffner (J.J) et Hodgins (J.K), "Synthesizing animations of human manipulation tasks".*Proceedings in SIGGRAPH2004*, pp 532-539, 2004.
- [42] LEE.J, ANDS HIN.S.Y, "A hierarchical approach to interactive motion editing for human-like figures". In *Proceedings of SIGGRAPH* pages 39-48.1999.
- [43] WELMANC, "Inverse Kinematics and Geometric Constraints for Articulated Figure Manipulation". Master's thesis, Simon Frasier University, 1993.
- [44] WOLOVICHW (A), ELLIOTH, "A computational technique for inverse kinematics ", *IEEE Conf. on Decision and Control*, pages. 1359-1363. 1984.
- [45] ARISTIDOU (A), LASENBYJ, "Fabrik : A fast iterative solver for the inverse kinematics problem. *Graphical Models*". Vol. 73, pages 243 - 260,2011.
- [46] HECKERC , RAABEB.,ENSLWR . W , DEWEESEJ . , MAYNARDJ , VANPROOIJENK. "Real-time motion retargeting to highly varied user-created morphologies". *ACM Transactions on Graphics (TOG)*. Vol. 27, pages 1-11 (2008).
- [47] UNZUETA L., PEINADO M., BOULIC R.,SUESCUNÃ, "Full-body performance animation with sequential inverse kinematics". *Graphical Models* pages 87-104,2008.
- [48] A. Witkin & Z. Popovi'c, "Motion warping". *Proceedings of ACM SIGGRAPH'95*, pages 105-107, 1995.
- [49] SENTISL, KHATIBO, "Synthesis of whole-body behaviors through hierarchical control of behavioural primitives" .*International Journal of Humanoid Robotics*.Vol. 2, (2005).
- [50] G. Dumont, S. Gibet , D. Gillard, J.-P. Jessel, F. Multon, "Mouvement et contrôle de mouvement".
- [51] Ph.FUCHS, G.MOREAU, Jean-Marie BURKHARDT, Sabine COQUILLART, "L'interfaçage, l'immersion et l'interaction en environnement virtuel", *Le traité de la réalité virtuelle - Volume 2*, Ecole des MINES de Paris, 2006.

- [52] I.AKLI, "Elaboration d'une stratégie de coordination de mouvements pour un manipulateur mobile redondant", mémoire présenté pour l'obtention du diplôme de magister, Université U.S.T.H.B, 2007.
- [53] R.GOURDEAU, "ROBOOP A Robotics Object Oriented Package in C++", Département de génie électrique Ecole Polytechnique de Montréal, 2007.
- [54] N.COURTY, E.ARNAUD, "Inverse Kinematics using Sequential Monte Carlo Methods", Université Joseph Fourier, INRIA Rhône-Alpes, France, 2008.
- [55] L.-C. T. Wang & C.Chen, "A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators". IEEE Transactions on Robotics and Applications, vol. 7, no. 4, pages 489–499, Août 1991.
- [56] J. Zhao & N. I. Badler, "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures". ACM Transactions on Graphics, vol. 13, no. 4, pages 313–336, 1994.
- [57] Wang, C.Chen, "A Combined Optimization Method for Solving the Inverse Kinematics Problem of Mechanical Manipulators". IEEE Transactions on Robotics and Applications, vol. 7, pages 489–499, 1991.
- [58] <https://fr.wikipedia.org/wiki/Humanode>.
- [59] H-ANIM. Human animation Working Group. <http://www.h-anim.org>.
- [60] https://fr.wikipedia.org/wiki/Cinématique_inverse
- [61] https://fr.wikipedia.org/wiki/Surface_implicite.
- [65] D. G. Luenberger. "Linear and non linear programming". Addison-Wesley, 1984.
- [66] <https://msdn.microsoft.com>
- [67] <https://www.autodesk.fr/products/3ds-max/overview>
- [68] <https://fr.wikipedia.org/wiki/OGRE>
- [69] <http://www.ogre3d.org/>
- [70] William Rowan Hamilton, "On quaternions and the rotation of a solid body". In Proceedings of the Royal Irish Academy, volume 4, pages 38–56, 1850.