

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ MOHAMED KHIDER - BISKRA
FACULTÉ DES SCIENCES EXACTES ET DES SCIENCES DE LA NATURE ET DE LA VIE
DÉPARTEMENT D'INFORMATIQUE



Thèse

présentée pour l'obtention du diplôme de

Docteur en troisième cycle : Informatique

spécialité

Image et vie artificielle

Édition interactive de mouvement pour le placement des caméras

Présentée par : Ahlem BENTRAH

Jury :

- Pr. Djedi Noureddine	Université de Biskra	Président
- Dr. Babahenini Mohamed Chaouki	Université de Biskra	Rapporteur
- Pr. Taleb ahmed Abdelmalik	Université de Valenciennes	Co-rapporteur
- Pr. Cherif Foudil	Université de Biskra	Examineur
- Pr. Moussaoui Abdelouahab	Université de Setif	Examineur

Année Universitaire 2014/2015

Dédicaces

À

ma grand mère,

mes chers parents Souad et Fateh,

ma sœur Chahde,

mes frères Fouad et Sami,

mon mari,

je dédie ce modeste travail.

Remerciements

Je tiens premièrement à prosterner remerciant Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail.

Je souhaiterais tout d'abord exprimer ma très profonde reconnaissance à Dr. Babahe-nini Mohamed Chaouki, mon directeur de thèse, pour son implication à la réalisation de ce travail de recherche et pour le support qu'il m'a apporté, sa patience, sa disponibilité et la pertinence de ses conseils m'ont été d'une aide précieuse tout au long de ce travail.

Je voudrais tout particulièrement remercier Pr.Philippe Pudlo pour m'avoir consacré du temps tout au long de ces années, ainsi que pour ses conseils et encouragements et pour son accueil et sa supervision lors de mon séjour en France.

Je voudrais remercier très chaleureusement Pr. Taleb Ahmed Adelmalik qui fait preuve de sa gentillesse et patience, et pour son accueil distingué au sein du laboratoire LAMIH, sans l'oublie de remercie le directeur et tous les membres du laboratoire LAMIH pour leur soutien durant mes stages de recherche au sein de leur laboratoire.

Je remercie également Dr. Djefal Abdelhamid pour son soutien constant dans ce manuscrit et pour son aide durant toute mes années du doctorat.

Je remercie le directeur du laboratoire LESIA, Dr. Cherif Foudil ainsi que Dr. Zitouni Othmane pour leur soutien durant toutes mes années d'étude.

Je remercie l'ensemble des membres de mon jury : Pr. Djeddi Nouredinne, Pr. Cherif Foudil et Pr. Moussaoui Abdelouahab pour l'intérêt qu'ils ont accordé à mon travail et pour avoir accepté de participer à ce jury.

Sur un plan plus personnel, j'aimerais dire un grand merci aux doctorants avec qui j'ai passé ces quatre dernières années. Mes amies Kalwach, Sara Kemlich, Abid Khadidja. Un remercie particulier à ma meilleure amie Nour Elhouda Benalia.

Enfin, je remercie mes parents et ma famille pour m'avoir encouragé pendant ces

nombreuses années passées, et pour avoir toujours été disponibles quand j'en avais besoin.

Résumé

La simulation des mouvements humains est importante dans de nombreuses applications, tel que le cinéma, la médecine, les jeux vidéo, le sport...etc. Les méthodes de capture de mouvement sont devenues très intéressantes pour générer des mouvements, au lieu de les produire à la main, l'idée sert à reproduire le mouvement d'un acteur réel. Cependant, ces mouvements capturés généralement ont des besoins spécifiques. Ainsi, la modification et la réutilisation de mouvement pour d'autres personnages est devenue un des domaines de recherche en plein développement, qui se connaît sous le nom d'édition de mouvement. Grâce à cet développement, nous comptons d'exploiter ce domaine pour le positionnement optimal des caméras dans les systèmes optiques de capture des mouvements.

Le but de cette thèse est de fournir aux utilisateurs un outil interactif pour les aider à modifier facilement et rapidement des mouvements préexistants. Dans un premier temps, nous présentons notre méthode de cinématique inverse utilisée pour appliquer des contraintes cinématiques à chaque frame, elle propose une solution au problème de conflits entre les tâches simultanées et évite les collisions entre les jointures de la posture finale et les objets de l'environnement.

Nous proposons ensuite un système de modification interactive des mouvements à base de la méthode de la cinématique inverse proposée permettant de modifier et adapter un mouvement à d'autres personnages d'une manière facile et rapide. Pour la modification des mouvements, le système permet aux utilisateurs de définir des contraintes d'une façon interactive et éditer le mouvement initial pour obtenir d'autres mouvements. Pour l'adaptation de mouvement, il permet à l'utilisateur d'appliquer des mouvements sur différents personnages qui n'ont ni la même géométrie ni la même topologie.

Les mots clés : Adaptation du mouvement, Cinématique inverse, Édition du mouvement.

ملخص

حركة الإنسان مهمة جدا في العديد من المجالات مثل الرياضة, الطب, الألعاب والسينما....الخ. تعتبر طرق التقاط الحركة من الطرق المهمة المستعملة في توليد حركات جديدة, بدلا من توليدها يدويا. تركز الفكرة على التقاط حركة شخص حي وتطبيقها على شخصية افتراضية. لكن, الحركة الملتقطة تتطلب احتياجات خاصة. لذلك, تعديل وإعادة استخدام الحركة لشخصية أخرى أصبح ميدان بحث في تقدم المعروف باسم " تحرير الحركة ". نظرا لهذا التقدم نحضر لاستغلال هذا المجال في مشكلة تحديد المواقع الكاميرا في نظام التقاط الحركة.

الهدف من هذه الرسالة هو أن نقدم لمستخدمي نظام التقاط الحركة, أداة تفاعلية لمساعدتهم على تعديل حركة موجودة من قبل بسهولة وبسرعة. أولا, قمنا بتطوير طريقة الكينيماتيك العكسية من اجل تغيير حركة الشخصية, هذه الطريقة تتغلب على مشكل النزاع بين المهام المتزامنة وذلك بعرض كيفيتين . اقترحنا أيضا طريقة للابتعاد عن العراقيل الموجودة في المكان

بعد ذلك؛ نقترح نظام تفاعلي لتعديل الحركات بالاعتماد على طريقة الكينيماتيك العكسية المقترحة مما يسمح بتعديل وتكييف الحركة لشخصيات أخرى بسرعة وسهولة. لتعديل الحركة، النظام يسمح للمستخدمين بتحديد القيود وتحرير الحركة الأولية للحصول على حركة أخرى. للتكيف مع الحركة، نقدم للمستخدم لتطبيق الاقتراحات لشخصيات مختلفة حيث والفرق هو في عدد المفاصل و طول أطراف الجسم.

الكلمات المفتاحية : الكينيماتيك العكسية , تغيير الحركات , تكييف الحركات

Abstract

The human motion is important in many applications as in the cinema, medicine, games and sport...etc. The motion capture methods became very interesting methods to generate new motions, instead of creating motion manually ; the idea is to reproduce the motion of an real actor. However, the captured motion require specific needs. Therefore, modifying and reusing the motion for other character became increasing research field known as "motion editing". Due to this development, we attend to exploit this field in the problem of positioning the camera in the motion capture system.

The aim of this thesis is to provide to the motion capture users, an interactive tool for helping to easily and rapidly modify preexisting movement. We first, we present our inverse kinematic solver used to apply kinematic constraints in each frame, it propose a solution to the conflict when dealing with many tasks simultaneously and avoid obstacle collision. Afterward ; we propose interactive system to modify a movements based on the proposed inverse kinematic solver allowing to modify and adapt a motion to other characters rapidly and easily. For the motion modification, the system allow to the users to define the constraint and edit an initial motion to obtain another one. For the motion adaptation, it offer to the user to apply a motions for different characters where the difference is in the topology and the geometry.

Key words : Inverse kinematic, Motion editing, Motion retargeting.

Publications et Communications

Les publications et les communications relatives à cette thèse sont les suivantes :

Publications

1. **Chapter Book** : A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet, P.Pudlo, A.Taleb-Ahmed. Full body adjustment using iterative inverse kinematic and body parts correlation. In Computational Science and Its Applications ICCSA 2014 (pp. 681-694). Springer International Publishing.
2. A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet, P.Pudlo, A.Taleb-Ahmed. Full Body Adjustment Using Iterative Inverse Kinematic and Body Parts Correlation. J. Mobile Multimedia 10(3-4) : 309-326 (2014).

Communications

1. A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet, P.Pudlo, A.Taleb-Ahmed. Online Motion Retargeting Method for Topologically Different Characters based Inverse Kinematic. 1ère Conférence sur l'Ingénierie Informatique (C2i).Alger 16-17 Décembre 2014.
2. A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet, P.Pudlo, A.Taleb-Ahmed. Interactive System Design of Scene for the positioning of cameras for Motion Capture ? La Journée algérienne de la Biomécanique et Innovation Oran. 19 Mai 2014
3. A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet,P.Pudlo, A.Taleb-Ahmed. Édition de mouvement utilisant cinématique inverse FABRIK étendue. Journée doctorial, 5 Décembre 2013, Valenciennes

Workshops

1. A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet,P.Pudlo, A.Taleb-Ahmed.Edition de mouvement en utilisant la cinématique inverse. WORKSHOP Images, Graphiques and Artificial life.Biskra 08,09-10 Juin, 2013.(poster+ présentation oral).
2. A.Bentrah, A.Djeffal, Med.Babahenini, C.Gillet,P.Pudlo, A.Taleb-Ahmed. Interactive System Design of Scene for the positioning of cameras for Motion Capture. Workshop on Artificial Intelligence Information and Communication Technologies. Biskra 2013. poster

Table des matières

Table des matières	I
Table des figures	IV
Liste des tableaux	VII
Introduction générale	1
1. Cadre de la thèse	2
2. Problématique de la thèse	4
3. Contributions	6
4. Organisation de la thèse	7
1 Édition de mouvement	9
1.1 Introduction	9
1.2 Représentation des personnages virtuels	10
1.2.1 Représentation squelettique	10
1.2.2 Représentation physique	11
1.3 Édition de mouvement	12
1.4 Caractéristiques des mouvements	13
1.5 Représentation du mouvement	13
1.5.1 Images clés ou postures	14
1.5.2 Segments ou primitives de mouvement	14
1.5.3 Représentations à dimension réduite	14
1.5.4 Données annotées	14
1.6 Représentation des rotations	15

TABLE DES MATIÈRES

1.6.1	Rotations d'Euler	15
1.6.2	Matrice de rotation	16
1.6.3	Quaternion	17
1.7	Méthodes d'édition de mouvement	18
1.7.1	Modification de mouvement	20
1.7.2	Combinaison de mouvements	32
1.8	Bilan	33
1.9	Conclusion	34
2	Cinématique Inverse	35
2.1	Introduction	35
2.2	Vocabulaires de la cinématique	35
2.2.1	Chaîne cinématique	36
2.2.2	Degré de liberté	36
2.3	Cinématique directe	37
2.4	Cinématique inverse	39
2.5	Problèmes de la cinématique inverse	41
2.5.1	Traitement de multiples contraintes et résolution de problème de conflit	41
2.5.2	Traitement de Problème de sous contraintes et de sur contraintes	43
2.5.3	Limites angulaires	44
2.5.4	Obstacles	45
2.6	Techniques de la cinématique inverse	46
2.6.1	Méthode analytique	46
2.6.2	Méthodes à base d'optimisation	48
2.6.3	Méthodes à base d'heuristiques	50
2.7	Bilan	55
2.8	Conclusion	57
3	PIKOV : Cinématique inverse avec évitement d'obstacles et multiples contraintes avec priorité	58
3.1	Introduction	58
3.2	Problématique	58
3.3	PIKOV : <i>Priority Inverse Kinematic with Obstacle Avoidance</i>	59
3.3.1	Évitement d'obstacles	60

3.3.2	Multiples tâches	63
3.3.3	Limites angulaires	66
3.4	Résultats et comparaison	66
3.4.1	Comparaison avec FABRIK	66
3.4.2	Comparaison avec PIK	69
3.5	Conclusion et discussion	70
4	PIKOV pour la Modification de trajectoire	72
4.1	Introduction	72
4.2	Aperçu général du système d'édition des mouvements	73
4.2.1	Adaptation du mouvement à une autre morphologie	73
4.2.2	Modification de trajectoire	74
4.3	Modification de trajectoire	74
4.3.1	Interactivité	74
4.3.2	Synergies	75
4.4	État de l'art sur les méthodes interactive d'édition de mouvement	75
4.5	Proposition	78
4.6	Aperçu de la méthode proposée	80
4.7	Contraintes	81
4.7.1	Contraintes des images clés	82
4.7.2	Contraintes de trajectoire clé	82
4.8	Ajustement utilisant la cinématique inverse PIKOV	83
4.8.1	Modèle du personnage	83
4.8.2	Ajustement du corps du personnage	87
4.9	Continuité des résultats	91
4.10	Résultats	92
4.11	Comparaison avec d'autres méthodes	95
4.12	Conclusion	96
5	Adaptation des mouvements à une autre morphologie utilisant PIKOV	98
5.1	Introduction	98
5.2	Adaptation d'un mouvement à une autre morphologie	98
5.3	Bilan et proposition	102
5.4	Aperçu du système	103
5.5	Adaptation des mouvements utilisant la cinématique inverse	104

TABLE DES MATIÈRES

5.5.1	Structure du fichier de mouvement	104
5.5.2	Mise en correspondance	105
5.5.3	Adaptation du tronc et du cou	107
5.5.4	Adaptation de la racine	108
5.5.5	Adaptation des jointures	110
5.6	Résultats	110
5.6.1	Adaptation de la racine	110
5.6.2	Respect des contraintes	113
5.7	Conclusion	116
Conclusion générale		117
Annexe : Architecture logicielle du système proposé		120
1	Introduction	120
2	Étape de création de l'outil	121
3	Architecture globale du système proposé	122
3.1	Moyens logiciels	122
3.2	Création et positionnement des caméras	122
3.3	Marqueurs	124
3.4	Mouvement	126
3.5	Création de la Base de mouvement	128
4	Conclusion	130
Bibliographie		131

Table des figures

1	Capture de mouvement dans un système VICON	1
2	Détection des positions des marqueurs dans un système VICON	2
3	Architecture de système pour la résolution de problème de positionnement des caméras	3
1.1	Représentation de squelette	11
1.2	Classification des méthodes d'édition de mouvement	20
1.3	Modification des basse fréquences Bruderlin et Williams [19]	21
1.4	Course avec émotions [101]	22
1.5	Système de modification de mouvement à base de contraintes [67]	25
1.6	Importance associée aux extrémités en fonction de leur proximité avec un objet	26
1.7	Adaptation des angles de jointures d'un bras en utilisant un squelette in- termédiaire [75]	27
1.8	Décomposition les trajectoires articulaires en splines hiérarchiques [67] . . .	27
1.9	Génération de mouvements naturels à base des contraintes spatio-temporelle dynamiques.[69]	29
1.10	Modification de mouvement à base des contraintes physiques [119]	31
2.1	Chaîne cinématique simple avec une racine et une seule extrémité	36
2.2	Types de jointure	37
2.3	Principe de la cinématique directe	38
2.4	Cinématique inverse	40
2.5	Principe de la cinématique inverse	40
2.6	Configuration simple	42

TABLE DES FIGURES

2.7	Positionnement d'un personnage dans une chaise en utilisant la stratégie de pondération	42
2.8	Le problème n'a pas de solution	43
2.9	Le problème a une seule solution	43
2.10	Le problème a deux solutions	44
2.11	le problème a une infinité de solutions	44
2.12	L'extrémité atteindre l'objet mais sans prendre en considération l'obstacle.	46
2.13	Résultats de la méthode SIK	47
2.14	Itération complète de FABRIK [5]	52
2.15	Résultats de SGPLVM pour la marche [37].	53
3.1	Stratégie d'évitement d'obstacles	60
3.2	Évitement d'obstacles en utilisant l'algèbre géométrique (GA)	63
3.3	Atteint de deux contraintes simultanément	65
3.4	Cas de multiple tâches	67
3.5	Atteint de trois contraintes simultanément	68
3.6	Évitement d'obstacles	69
4.1	Vue globale du système proposé d'édition des mouvements	73
4.2	Génération des mouvements avec un modèle déformable de mouvement	76
4.3	Édition de mouvement de la marche [82]	77
4.4	Processus d'ajustement d'une posture	79
4.5	System overview	80
4.6	Exemple de contraintes spécifiées par l'utilisateur	82
4.7	Hierarchie de personnage virtuel	84
4.8	Modèle du personnage : hiérarchie proposée	84
4.9	Modèle du personnage : Degré de liberté	85
4.10	Représentation du personnage : découpage de personnage en groupe	86
4.11	Préhension d'un cible avec une chaine simple	88
4.12	Cas d'inclination du tronc dans le mouvement	89
4.13	Propagation des modifications	92
4.14	Atteindre un objet seulement avec le bras	93
4.15	Atteindre un objet utilisant le tronc et le bras	93
4.16	Multiple cible à atteindre	94
4.17	Quatre cibles à atteindre	94

TABLE DES FIGURES

5.1	Problème de recopie directement d'un mouvement sur un personnage différent	99
5.2	Adaptation de mouvement aux d'autres morphologies [42]	99
5.3	Adaptation du mouvement utilisant un squelette indépendant.[62]	101
5.4	Adaptation de mouvement utilisant un squelette intermédiaire.[75]	102
5.5	Adaptation de mouvement à une nouvelle physionomie[57].	102
5.6	Schéma général de système d'adaptation des mouvements	104
5.7	Deux squelettes avec différents tronc	106
5.8	Renommer les jointures communes entre le fichier de mouvement et le personnage virtuel	106
5.9	Adaptation du tronc	108
5.10	Problème souvenue lors de recopie directement la position de la racine	109
5.11	Test1 : Adaptation de la racine pour un personnage plus grands que le personnage source	111
5.12	Tes2 : Adaptation de la racine pour un personnage plus petit que le personnage source	111
5.13	Trois personnes utilisés pour adapter le mouvement de la marche	112
5.14	Application du même mouvement de la marche sur les trois personnes	112
5.15	Adaptation de la racine des personnage pour un mouvement de la marche	113
5.16	Mouvement de préhension par personnages en les mêmes caractéristiques	113
5.17	Mouvement de préhension par des personnages n'ont pas la même géométrie	114
5.18	Application seulement l'étape d'adaptation de la racine	115
5.19	application de la méthode PIKOV afin de préserver les contraintes de position, le personnage atteindre l'objet	115
5.20	Graphe de comparaison entre la trajectoire en x,y de la main de personnage d'origine et celle de personnage de destination	116
21	Étape de création de l'outil	121
22	Architecture globale du système proposé	122
23	Interface illustrant les opérations que peut effectuer l'utilisateur sur une caméra.	123
24	Représentation des caméras dans le système.	124
25	Représentation des marqueurs	126
26	Interface sert à importer des fichiers	127
27	Résultat de l'importation de fichier du mouvement	128
28	Création de la base du mouvement	129

Liste des tableaux

2.1	Comparaisons des principales méthodes de cinématique inverse	56
2.2	Caractéristiques des méthodes principales d'évitement d'obstacle	57
3.1	Comparaison différentes méthodes de cinématique inverse	70
4.1	Comparaison entre des méthodes de modification des mouvements	78
4.2	Comparaison entre une posture capturé et une autre construite par la méthode proposée	95
5.1	Comparaison entre les dimensions du personnage virtuel	112
2	Caractéristiques des marqueurs dans notre outil	124

Introduction générale

Actuellement, l'utilisation des systèmes de capture de mouvement connaît une très grande évolution dans plusieurs domaines d'application tels que la médecine, l'industrie des jeux vidéo, l'industrie cinématique, ...etc. En effet, plusieurs technologies de capture de mouvement existent, certaines sont mécaniques, d'autres magnétiques ou inertielle, mais la plus utilisée est la technologie optique pour des raisons de flexibilité des mouvements de sujet .

La technologie optique utilise généralement deux familles de systèmes : les systèmes basés sur les images ou la vidéo et les systèmes basés sur l'utilisation de caméras spécifiques et des marqueurs fixés aux articulations des objets. Les caméras spécifiques et les marqueurs associés permettent de localiser facilement les éléments caractérisant l'objet et calculer sa position 3D et suivre ainsi son mouvement. Le système VICON en est un exemple de ce type de systèmes. La figure 1 suivante représente une application utilisant le système VICON.

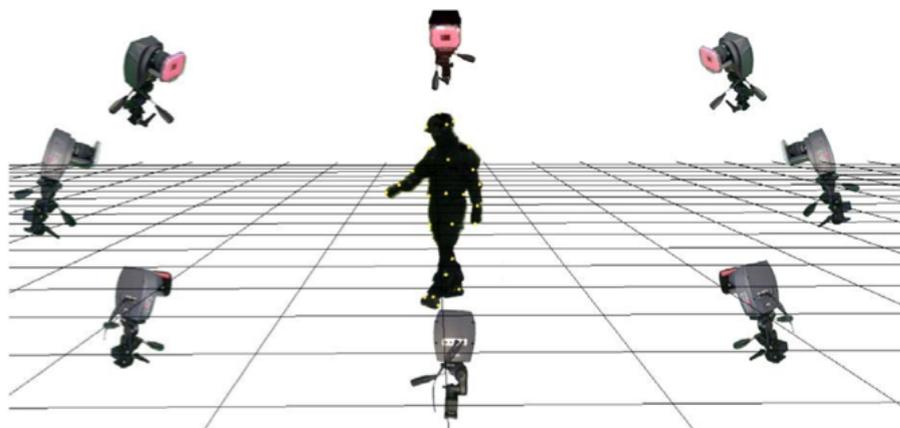


FIGURE 1 – Capture de mouvement dans un système VICON

Dans un tel système les caméras sont reliées à travers un dispositif (hub) à un ordinateur contenant un software permettant de récupérer les positions 3D et reconstruire la scène réelle en scène virtuelle (Figure 2).

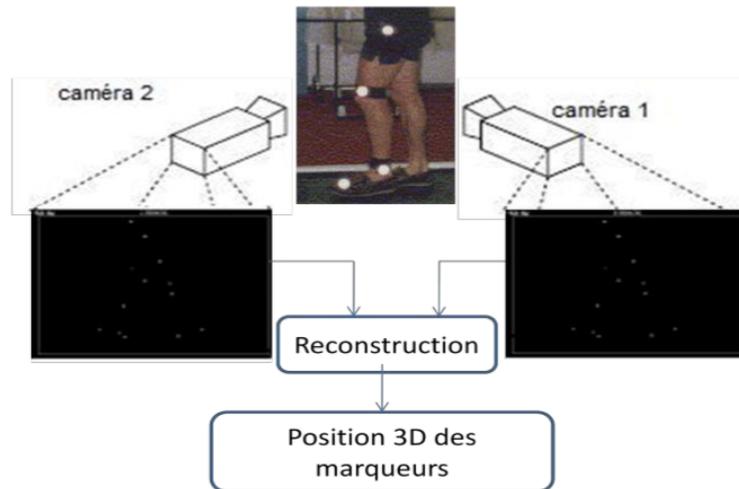


FIGURE 2 – Détection des positions des marqueurs dans un système VICON

1. Cadre de la thèse

Cette thèse s'inscrit dans le cadre d'un projet Tassili entre le laboratoire LAMIH (Laboratoire d'Automatique, de Mécanique et d'Informatique industrielles et Humaines) de l'université de Valenciennes et le laboratoire LESIA (Laboratoire de Recherche des Systèmes Experts, Imagerie et leurs Applications dans l'ingénierie) de l'université de Biskra. Le projet traite la problématique de positionnement automatique des caméras dans un système VICON. En effet, le problème est dû aux points suivants :

1. La position 3D d'un marqueur ne peut être détectée par une seule caméra,
2. L'utilisation de deux caméras ne suffit pas, à cause des éventuelles occultations des marqueurs par l'objet lui même ou par les obstacles qui peuvent exister dans la scène, ou même par d'autres marqueurs.
3. La précision des positions des marqueurs augmente avec l'augmentation des caméras,
4. Les positions des caméras dans la scène sont des paramètres crucial pour d'une part capter avec précision tous les marqueurs, et d'une autre par optimiser le nombre de caméras.
5. Actuellement, les caméras sont positionnées dans la scène par un opérateur humain

1. CADRE DE LA THÈSE

expérimenté, ce qui représente un travail fastidieux et de faible précision, aussi l'utilisation d'un nombre élevé de caméras.

Le projet proposé vise à développer un logiciel qui prend en entrée une scène virtuelle, simulant la scène réelle, un personnage virtuel simulant le sujet réel, et le mouvement souhaité du sujet et produit en sortie le nombre optimum de caméras nécessaires à la capture du mouvement de l'objet réel ainsi que leur positions optimales.

Pour résoudre cette problématique le travail a été repartitionné en trois modules (Figure 3) :

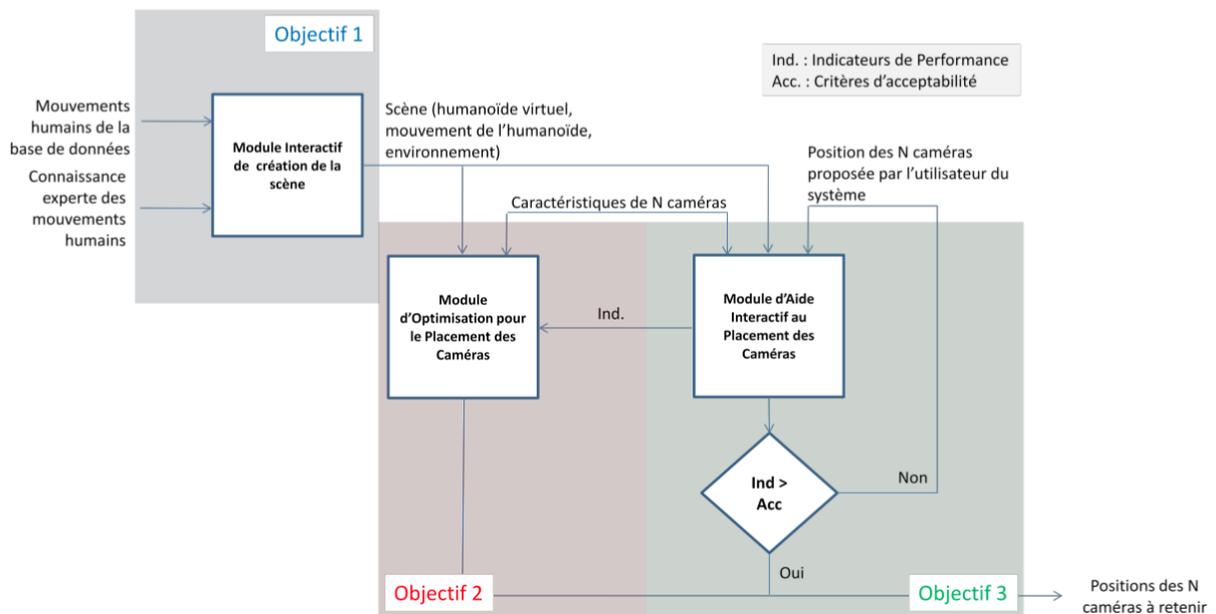


FIGURE 3 – Architecture de système pour la résolution de problème de positionnement des caméras

1. **Module interactif de création de la scène** : qui vise à développer un outil de création de scène en vue de positionnement de caméra. Cette partie consiste à :
 - (a) Modéliser la salle d'expérimentation en prenant en compte les mesures réelles de la salle.
 - (b) Modéliser les objets de la scène, par exemple un humanoïde virtuel avec sa structure, ses degrés de liberté et lui attacher les marqueurs ,
 - (c) Attribuer le mouvement à l'objet. Les mouvements peuvent être extraits d'une base de mouvement déjà capturés et adaptés ou créés à l'aide du système,
 - (d) Définir un ensemble de caméras avec leurs caractéristiques et les placés dans la scène.

2. **Module d'optimisation pour le placement des caméras** : Les informations sur l'objet, son mouvement et l'environnement sont fournis comme entrées à ce module qui les utilisent avec des indicateurs de performance et des algorithmes d'optimisation adéquats pour fournir un positionnement optimal des caméras.
3. **Module d'Aide Interactif au Positionnement des Caméras** : Ce module sert à optimiser le module précédent en lui proposant des positions initiales permettant sa convergence. Ces positions peuvent être extraites d'une base de cas après une étude de similarité.

Cette thèse s'intéresse au premier module, visant la création de la scène virtuelle avec tous ses composants à savoir la salle, les obstacles, les caméras, l'humanoïde virtuel, ses marqueurs et enfin la création et l'édition de son mouvement.

2. Problématique de la thèse

Pour atteindre l'objectif de notre module ; nous nous proposons de traiter les points suivants :

- Modélisation d'un humanoïde virtuel paramétrable,
- Définition du contenu de fichier de mouvement,
- Adaptation de mouvement aux différentes morphologies,
- Modification interactive du mouvement d'une façon simple.

Dans l'architecture de l'outil proposé deux aspects sont présentés : développement de logiciel et recherche scientifique. Il existe donc des modules de programmation pure comme la modélisation et la position des caméras virtuelles, comme il existe deux modules qui représentent des domaines de recherche actives. La modélisation de l'humanoïde et la création /modification des mouvements.

Pour la modélisation des humains virtuels, il existe actuellement certain nombre de représentations standards du personnage virtuel. Le niveau de détail du modèle de l'humain virtuel dépend de l'application, dans les applications médicales et biomécanique, on utilise des modèles complexes avec plus de détails, par ailleurs, les modèles utilisés dans le domaine de l'animation sont généralement dénués de connaissance anatomiques, ce qui donne des mouvements peu naturels.

La simulation des mouvements des humains virtuels est un domaine de recherche très actif. De nombreux travaux de recherche s'intéressent à la génération des mouvements naturels en développant de nouvelles méthodes d'animation afin de contrôler l'humain virtuel

dans les environnements complexes. Pour ce faire, on prend en compte des contraintes cinématiques gérant les positions et les rotations des jointures du personnage, des contraintes de style qui prennent en compte la morphologie ou les maladies pour améliorer la qualité des résultats, aussi, des contraintes dynamiques. Cependant, il faut assurer d'une part que l'algorithme soit capable de générer des mouvements naturels et d'autre part d'obtenir des résultats en temps convenable.

Globalement, la solution proposée consiste à réutiliser le mouvement d'une manière intelligente. L'avantage majeur de ce type de méthodes est que les résultats de la simulation sont basés sur des mouvements naturels mesurés sur des acteurs réels ce qui donne des mouvements naturels, leur problème consiste à adapter le mouvement de base afin de répondre à toutes les situations pouvant se produire durant l'application. En effet, il est impossible de capturer tous les mouvements possibles pour tous les humains virtuels. Ce problème est compensé par des algorithmes de modification des mouvements afin de répondre à différentes situations, ce qui représente une tâche difficile pour deux raisons : Premièrement, les humains sont naturellement experts lorsqu'il s'agit de voir les autres humains. Cependant, toute inexactitude subtile dans le mouvement de l'humanoïde virtuel conduit à des sentiments étranges pour les observateurs humains. On ne sait pas expliquer clairement le problème, mais on sait que le mouvement n'est pas naturel. Deuxièmement, le corps de l'humain a une architecture complexe contenant plusieurs degrés de liberté (DDL), et comme il est impossible de changer les paramètres de chaque articulation à la main, des algorithmes doivent être développés afin de faciliter l'animation de ces structures.

Pour résoudre les problèmes sus cités, plusieurs chercheurs utilisent la cinématique inverse (IK), qui donne la possibilité à l'utilisateur de modifier des mouvements en attachant des contraintes sur le corps du personnage virtuel puis glisser les jointures aux nouvelles positions, ensuite, un solveur de la cinématique inverse calcule la nouvelle posture.

La cinématique inverse représente le cœur des techniques d'édition de mouvement à base des contraintes tels que dans le travail de Lee et Shin [66], Gleicher [35], Liu et al [71] et Callenec and Boulic [64], vu qu'elle fournit une édition de mouvement et qu'elle permet à l'utilisateur de spécifier les contraintes sur tout le mouvement ou sur des parties précises.

Cette thèse étudie le problème d'édition de mouvement pour le positionnement des caméras des systèmes optiques de capture de mouvements.

3. Contributions

Malgré l'existence de plusieurs méthodes récentes d'édition de mouvement, elles restent insuffisantes à cause de plusieurs raisons :

- Les méthodes récentes d'édition de mouvements exigent l'existence préalable d'une base de mouvements, qui est généralement difficile à obtenir.
- Elles nécessitent une étape de pré-traitement de la base de mouvement ce qui rends la méthode hors ligne.
- Les mouvements enregistrés dans la base sont toujours liés aux caractéristiques et type du personnage considéré, ce qui limite les résultats de l'édition.

Dans notre thèse nous voulons proposer une méthode d'édition de mouvement afin de l'utiliser pour simuler un système de capture de mouvement dans un laboratoire de recherche dont :

- Il existe différents types de sujets (mobilité réduite, personnages âgées, sportifs ?.) et différent type de mouvement ce qui complique l'obtention d'une base pour chaque sujet et pour chaque mouvement
- Le Mouvement d'entrée est déjà similaire au mouvement à capturer, pour ce raison nous avons pas s besoin de créer un autre mouvement, il suffit de faire des modifications légères du mouvement d'entrée.
- La méthode à proposer doit être facile à utiliser par les novice dont définition des contraintes doit être simple , pas besoin d'entrer dans les détails mathématiques ou physiques.

Pour ces raisons, notre première contribution est la proposition d'une méthode de cinématique inverse PIKOV (Cinématique inverse avec priorité et évitement d'obstacles) qui n'utilise pas une base de mouvement. La deuxième contribution consiste à utiliser la méthode de la cinématique inverse proposée pour l'édition des postures. Elle permet de traiter plusieurs contraintes simultanément sans conflit, ainsi que de générer des postures sans collision avec d'autres objets de l'environnement. C'est une méthode rapide qui respecte la synergie entre les membres du corps du personnage virtuel. La cinématique inverse PIKOV est utilisée dans la troisième contribution qui concerne l'adaptation d'un mouvement d'une morphologie à une autre. La méthode proposée ajuste automatiquement la position de la racine par rapport à la géométrie du personnage d'origine ensuite respecte les caractéristiques essentielles de mouvement d'origine utilisant la cinématique inverse. Aussi, elle résout le problème des jointures de personnage de destination qui n'ont pas de

correspondance dans le fichier de mouvement. L'utilisation de de la méthode de PIKOV pour l'édition de mouvement garantie le compromis entre le réalisme de mouvement et la rapidité d'obtention de ce mouvement. Notre méthode d'édition e mouvement n'as pas besoin d'une base de mouvement

4. Organisation de la thèse

Cette thèse est structurée comme suit :

Dans le premier chapitre, nous donnons une classification des méthodes de modification de mouvement, les méthodes à base de la cinématique inverse sont les plus appropriées afin de modifier un mouvement en spécifiant des contraintes géométriques.

Dans Le deuxième chapitre, nous présentons les méthodes à base de la cinématique inverse en détail. La méthode FABRIK récemment proposée donne en particulier des résultats rapides ce qui répond aux exigences des applications de modification interactives, cependant son principe peut engendrer des conflits lors de la manipulation de plusieurs tâches en même temps. Au plus, les résultats peuvent être insatisfaisants dans le cas de la présence des obstacles.

Dans le troisième chapitre, nous présentons une méthode de cinématique inverse PIKOV qui a le même principe de base que la méthode FABRIK(Farward Backward Inverse Kinematic) : elle parcourt des chaines en avant et en arrière afin d'atteindre l'objectif. La méthode que nous proposons permet en outre de résoudre le problème de conflit lors du traitement des tâches simultanément ; de plus, elle prends en compte les obstacles situé dans l'environnement.

Dans le quatrième chapitre, Nous présentons notre technique de modification du mouvement à base de la cinématique inverse PIKOV, son principe de base consiste à découper le personnage virtuel en groupes, et ensuite à appliquer la cinématique à chaque groupe tout en prenant compte des relations entre les groupes.

Dans le cinquième chapitre, nous présentons notre technique d'adaptation de mouvement d'un personnage à une autre morphologie, cette méthode se base sur la cinématique inverse, elle sert à adapter le mouvement à un autre personnage qui n'a pas les mêmes dimensions des segments ni le même nombre de jointure.

Une conclusion des travaux présentés dans cette thèse ainsi que les perspectives de leur utilisation sont présentées dans le dernier chapitre.

Enfin, un annexe est consacré à la description de l'outil de modification du mouvement

4.ORGANISATION DE LA THÈSE

pour le positionnement de caméras où nous allons présenté chaque module de l'outil séparément.

Édition de mouvement

1.1 Introduction

Notre problématique est focalisée principalement sur la modification interactive des postures en temps réel pour simuler des mouvements envisager par l'utilisateur et trouver la position optimale des caméras afin de capturer ces mouvements.

Plusieurs techniques arrivent à modifier le mouvement d'un personnage virtuel en temps réel. Ces techniques diffèrent dans le compromis qu'elles offrent entre le contrôle qui peut être exercé sur le mouvement, la nature des résultats, et le temps de calcul nécessaire, le choix de la bonne technique dépend des exigences de l'application où l'on va appliquer. Ce chapitre fournit un aperçu sur les méthodes d'édition du mouvement utilisées afin de modifier un mouvement.

Afin de parvenir à animer ou modifier un mouvement d'un personnage virtuel, la première étape consiste à modéliser ce dernier. Il convient donc à faire des choix sur les parties et les articulations du corps qui sont pertinentes ou non dans le cadre de l'animation des personnages virtuels. Cette modélisation est discuté dans la section 1.2. Les notions liées aux mouvement tels que sa définition, ses caractéristiques et les différentes représentations des postures sont discuté dans les sections 1.4 ; 1.5. La section se propose 1.6 de décrire les différentes représentations des rotations au niveau du mouvement qui joue un rôle essentiel dans la qualité et la simplicité de modification. Enfin, dans la section 1.7, nous donnerons notre classification des méthodes utilisées pour l'édition des mouvements.

1.2 Représentation des personnages virtuels

Afin de parvenir à modifier le mouvement d'un personnage virtuel, la première étape consiste à le modéliser. Il convient donc de faire des choix sur les informations du corps humain utiles dans le cadre de l'édition des mouvements des personnages.

Plusieurs représentations de personnage virtuel sont proposées dans la littérature, elles dépendent du type de simulation : pour les simulations cinématiques, une représentation squelettique avec les degrés de liberté de chaque jointure suffit, par contre si la simulation est dynamique il est nécessaire d'ajouter des informations sur l'aspect physique du personnage virtuel comme les forces, les inerties....etc.

1.2.1 Représentation squelettique

Une façon simple de représenter un personnage virtuel consiste à définir sa squelette seulement à partir des positions de ses articulations. Johanson [49] a montré que cette représentation est suffisante pour percevoir les mouvements. Néanmoins, utilisant seulement cette information ne garantit pas que la distance entre les articulations est constante ce que peut conduire à une désarticulation du système. C'est la raison pour laquelle une représentation hiérarchique est utilisée. En effet, chaque os est représenté par un segment et les nœuds de la hiérarchie sont par conséquent les articulations qui rattachent les os entre eux. L'animation de personnage peut alors être faite en utilisant les trois degrés de liberté de chaque segment au cours du temps. Ainsi, la distance entre les articulations du squelette est conservée. Cependant, afin de faciliter l'animation du personnage virtuel, quelques simplifications géométriques sont utilisées, tel que :

- La représentation du squelette sous forme hiérarchique suppose le choix d'une articulation comme racine de l'arborescence, elle est généralement placée au pelvis. La racine est le seul nœud possédant trois degrés de liberté de translation en plus des trois de rotation. Ainsi, chaque jointure a un parent et éventuellement des fils. Les rotations et les translations des jointures du parent affectent toutes ses jointures fils. Les chemins de la racine à la dernière jointure dans le squelette sont appelés "sous chaînes".
- La complexité du squelette joue un rôle important pour pouvoir contrôler précisément l'ensemble des os de squelette. La manipulation d'un nombre limité d'articulation permet d'alléger la structure tout en conservant un niveau de détail suffisant pour retranscrire les principales caractéristiques d'un mouvement humain. La diminution

1.2. REPRÉSENTATION DES PERSONNAGES VIRTUELS

du nombre d'articulations 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 1.1 donne un aperçu d'une représentation du squelette utilisé dans notre travail. La figure à gauche montre un squelette simplifié sous une forme d'arborescente où la racine de squelette est le pelvis.

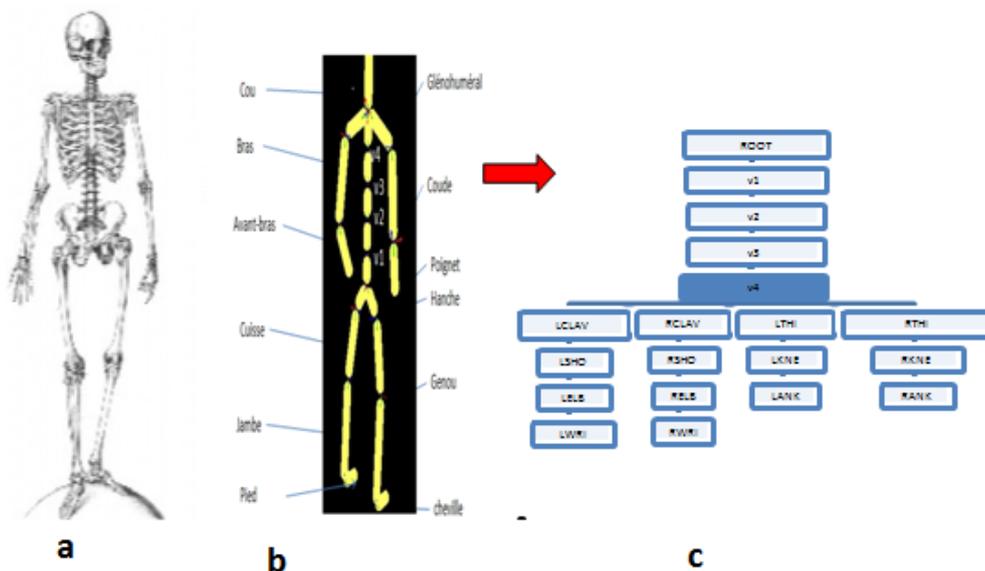


FIGURE 1.1 – Représentation de squelette. a : Squelette humain, b : Représentation simplifiée du squelette, c : Structure arborescente du squelette

L'application des mouvements créés pour un personnage virtuel à un autre personnage d'une façon facile nécessite la standardisation des topologies des personnages virtuels utilisées dans toutes les applications. Le standard H-anim [38] offre une standardisation complète des noms des jointures et ces topologies, spécifiant leurs positions de départ et la façon dont elles sont connecté.

1.2.2 Représentation physique

Pour la simulation physique, les personnages doivent contenir des attributs qui ne sont pas exigés dans la simulation à base de la cinématique.

Dans cette représentation, Les segments du personnage nécessitent à la fois la géométrie, qui est utilisée pour la détection des collisions, ainsi que les informations sur la

masse, qui sont utilisées durant la simulation physique. Les segments corporels sont modélisés en utilisant des formes primitives, comme des boîtes, des sphères ou des cylindres afin d’augmenter les performances de la détection des collisions. Ces formes sont utilisées afin de calculer l’information sur la masse des membres, en supposant que sa distribution est uniforme. Une autre façon de déduire les propriétés des masses est l’utilisation des données cadavériques [25].

1.3 Édition de mouvement

Il existe des techniques traditionnelles d’animation consistant à créer un mouvement, en utilisant : des images clés [63] qui sert à dessiner les postures par un animateur expérimenté, des méthodes procédurales[16] qui se fondent sur des algorithmes basé sur des fonctions variant au cours de temps pour contrôler les paramètres bas niveau du mouvement ou des méthodes qui utilisent des contrôleurs physiques [81] afin de créer le mouvement. Chaque méthode a ses avantages et ses limites, mais l’inconvénient commun entre ces techniques est la génération des mouvements moins réalistes que le mouvement d’un personnage réel.

Les méthodes d’animation traditionnelles sont remplacées par des méthodes d’animation à base de mouvements déjà capturés. A première vue, les mouvements fournis par les techniques de capture de mouvement sont correctes et ne nécessite aucune modification.

L’utilisation directe de tels mouvements capturés ne peut être que sous des contraintes tels que :

- Avoir une correspondance entre l’anatomie de la personne enregistrée et du personnage virtuel.
- Avoir le même environnement, lors de la capture et lors de l’animation.
- Réaliser la même tâche lors de la capture et lors de l’animation.

Ces conditions étant rarement toutes rassemblées, l’utilisation du mouvements capturés à l’état brute est très rare et généralement limitée aux études faites en sciences du mouvement.

Le besoin de l’édition du mouvement ne peut être nié, même si le mouvement est correcte, l’imperfection des systèmes de capture nous oblige à le corriger après son acquisition. En pratique, plusieurs besoins nous conduisent à utiliser l’édition des mouvements, nous pouvons les résumer comme suit :

- Rectification : qui consiste à faire des changements dans le mouvement capturé afin

de le corriger et éventuellement reconstruire le mouvement d'une façon plus précise [103].

- Réutilisation : les mouvements capturés sont naturellement liés à des situations spécifiques et leur réutilisation pour des situations un peu différente tel que la différence dans la morphologie [41].
- Changement d'objectif : malheureusement, on ne peut pas toujours prédire le type de mouvement dont on a besoin. Parfois on veut modifier ce mouvement après son acquisition [82, 64].

Au niveau de notre thèse, nous nous sommes intéressés à la réutilisation et le changement d'objectif d'un mouvement déjà capturé. Avant d'entamer les travaux existants dans ce domaine, nous allons définir quelques concepts que nous utiliserons par la suite.

1.4 Caractéristiques des mouvements

Avant de présenter les méthodes d'édition, nous devons définir ce que nous allons changer dans un mouvement. La notion des propriétés ou caractéristiques des mouvements est très importante pour l'édition des mouvements, elles permettent de définir les aspects à modifier ou à préserver lors de la modification d'un mouvement.

Il existe plusieurs niveaux des propriétés des mouvements qui doivent être pris en considération. Il y a des caractéristiques dites « bas niveau » [22] comme par exemple, dans le temps t_1 , la jambe est toute droite, dans le t_2 les pieds touchent le sol. .etc. Chacune de ces caractéristiques peut être observée dans le mouvement sans besoin d'une compréhension profonde du mouvement ou de contexte, elle possède une brève description mathématique. Aussi, des caractéristiques dites de « haut niveau » qui décrivent le style et l'état psychique de personnage tels que la joie, la tristesse, le colère...etc, de tel caractéristique sont très difficile à décrire [37].

1.5 Représentation du mouvement

Plusieurs représentation des séquences des mouvements sont proposé dans la littérature. Cette section propose donc de décrire, dans un premier temps, ces différentes représentations du mouvement.

1.5.1 Images clés ou postures

C'est la représentation la plus utilisée dans les systèmes de modification des mouvements. Les mouvements sont représentés par des vecteurs de postures de grande dimension. Afin de préserver l'information temporelle de mouvement, nous ajoutons des vecteurs de vitesse et d'accélération. Chaque posture étant elle-même un vecteur de position et de rotation [6].

1.5.2 Segments ou primitives de mouvement

Une autre représentation plus significative est nécessaire dans certaines applications. Elle consiste à segmenter temporellement les mouvements en clips de mouvements plus petits, chaque clip étant considéré comme une action de plus haut niveau, et utilisé pour composer des comportements plus complexes. Certains auteurs proposent de regrouper des actions similaires afin de constituer ce qu'ils appellent des primitives de mouvement [47]. En étudiant et modélisant la façon dont ces primitives sont combinées, il est ainsi possible de produire des séquences autonomes.

1.5.3 Représentations à dimension réduite

Pour des mouvements représentés par des postures, l'ensemble des données d'apprentissage peuvent atteindre de grandes dimensions. Cela s'aggrave lorsque les vitesses et les accélérations sont incluses, la dimension des données étant alors multipliée par trois. En travaillant dans ces espaces de grande dimension, il devient difficile de comprendre et de visualiser toute opération appliquée sur les données. Cela implique également un temps de calcul et des ressources machine accrues. Les chercheurs ont proposé diverses techniques de réduction des dimensions pour traiter les séries chronologiques multi-attributs [22, 90].

1.5.4 Données annotées

Dans leur forme brute, les informations de mouvement manquent de signification ; à partir des postures, nous pouvons pas dire si un mouvement de marche, de préhension ou de danse sont celles d'un homme, d'une femme ou d'un enfant.

Une façon classique d'ajouter du sens aux mouvements est de les annoter avec des informations d'ordre sémantique (acteurs, styles, émotions, etc.). Cette annotation guide le synthèse de nouveaux mouvement [4].

L'annotation des données de mouvement est habituellement un processus manuel, [24] a annoté les périodes de contact entre les pieds et le sol tout au long du mouvement afin de adapter un mouvement à une autre morphologie. Cependant, certains travaux [60, 45] permettent d'effectuer l'annotation automatique des données.

Après cette petite explication sur les différentes représentations des mouvements existantes, nous allons nous concentrer sur la représentation par posture, qui est très utilisée dans la littérature.

Après avoir vu les différentes représentations des mouvements possibles, nous abordons maintenant la manière de représenter les rotations des mouvements.

1.6 Représentation des rotations

Afin de représenter la configuration du personnage, nous devons avoir des informations sur la position d'un point dans le squelette, et l'orientation du squelette. L'orientation est exprimée comme la rotation relative à d'autres systèmes de coordonnées, soit fixe ou relative ceux qui nous donnent deux types d'orientations globales ou relatives. Les orientations globales sont par rapport au système de coordonnées fixe de la scène, les orientations relatives sont par rapport à un système de coordonnées relative à un objet ou à une autre jointure. Le passage d'un système de coordonnées à l'autre est simple. Les orientations relatives sont généralement utilisées dans le squelette afin d'exprimer les rotations entre deux articulations (jointures) reliées entre elles.

Pour quantifier ces transformations géométriques, plusieurs représentations mathématiques existent, chacune possédant ses propres avantages et inconvénients, sachant que le passage entre les représentations est possible. Nous introduisons dans cette section les plus couramment utilisées : les angles d'Euler, les matrices de rotation, quaternion et Exponential Maps.

1.6.1 Rotations d'Euler

Dans cette représentation, chaque rotation 3D dans l'espace peut être construite comme une séquence de rotations individuelles autour des axes. Les rotations sont décrites utilisant : yaw, roll et pitch

- Yaw : la rotation de gauche / droite autour de l'axe X
- Pitch : la rotation haut / bas selon l'axe Y.

– Roll : c'est la rotation tilt (inclinaisons) autour l'axe z.

Les angles Euler ont un avantage où n'importe quelle 3 valeurs représentent une rotation. Pour modifier un mouvement, il suffit de changer les valeurs qui représentent le mouvement selon nos besoins. La représentation par angles Euler peut être utile par ce qu'elle est supportée par plusieurs bibliothèques d'animation. Cependant, elle rencontre certaines problèmes. tels que la dépendance des résultats à l'ordre des opérations qui est très important : si nous appliquons un pitch sur un objet avec angle 90 degré, ensuite un yaw de 90 degrés, nous obtiendrons un résultat différent que celui si nous appliquons un yaw avant le pitch. Un autre problème de cette méthode est la perte d'un degré de liberté ou Gimbal Lock en anglais. Comme la position finale d'un objet dépend de l'ordre d'application des rotations suivant les 3 axes, il arrive que parfois l'une des rotations autour d'un axe se confond avec un autre axe de rotation ce qui entraîne l'impossibilité de tourner l'objet selon le troisième axe ainsi perdu (exemple séquence de rotation d'angles 90 degrés).

1.6.2 Matrice de rotation

Afin de représenter des rotations 3D, la majorité des API graphique utilisent les matrices de rotation. En effet, les matrices permettent de représenter et de combiner toutes les transformations : rotation, translation et même mise à l'échelle (cette dernière n'est pas utilisée dans l'animation des personnages virtuels) qui peuvent être appliquées à un vecteur de configuration (dans le cas de l'animation des personnages virtuels). Les matrices ont un grand avantage parce qu'elles unifient toutes les transformations dans l'espace. Néanmoins, les matrices de rotation se rapportent aux angles Euler, c'est pourquoi elles souffrent du même problème liée à ce dernier où sont soumises aux limitations liées aux angles d'Euler. De plus, les matrices de rotation sont extrêmement coûteuses en espace mémoire vue leur représentation par des tableaux bidimensionnelles de taille 3x3 ou 4x4.

Les matrices ci-dessus représentent respectivement les rotations $R_{(\alpha,x)}$, $R_{(\beta,y)}$ et $R_{(\gamma,z)}$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \sin\beta & 0 & \cos\beta \\ 0 & 1 & 0 \\ \cos\beta & 0 & -\sin\beta \end{pmatrix} \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = R_{(\alpha,x)} * R_{(\beta,y)} * R_{(\gamma,z)} \quad (1.1)$$

Où R représente la matrice de transformation qui combine les rotations selon les trois axes.

1.6.3 Quaternion

Comme nous avons déjà mentionné, les angles d'Euler ont quelques défauts dus à l'ordre dans lequel les rotations doivent être réalisées. Les quaternion sont proposé par proposés par Lord Williams Hamilton [39] Pour stocker une rotation dans un Quaternion, on utilise non plus une série de rotation successives autours de chaque axe comme dans les angles d'Euler mais un vecteur arbitraire et un angle de rotation quelconque autour de ce vecteur. Les quaternions sont des extensions des nombres complexes : La partie réelle du Quaternion contenant l'angle et la partie imaginaire le vecteur 3D. Le Quaternion étant obligatoirement unitaire, c'est à dire de norme 1.

$$quat = w + xi + yj + zk = (w; v) \quad (1.2)$$

w est la partie réelle, v la partie imaginaire.

Les quaternions offrent d'énormes avantages, le premier est la place mémoire relativement faible qu'ils occupent comparativement avec celle que nécessitent les matrices. Leur second avantage est la facilité qu'ils offrent pour la combinaison de rotations. Tout comme les matrices, les Quaternion de rotation se combinent par multiplication mais cette fois se sont bien des rotations arbitraires qui sont combinées et le nombre d'opérations nécessaires à cette combinaison est bien inférieur à celui requis pour combiner des matrices. Une telle économie de calculs est extrêmement importante lorsque nous travaillons par exemple avec la représentation d'objets hiérarchiques (l'animation d'un squelette) ou de cinématique inverse. De plus, la représentation angle-axe ne souffre pas du problème du Gimble Lock et l'utilisation des quaternions permet ainsi d'obtenir les résultats.

L'inconvénient majeur de l'utilisation de cette représentation pour l'édition de mouvement est que les paramètres ne sont pas intuitifs. La manipulation directe des quatre valeurs par un utilisateur est très difficile. La représentation par quaternion est souhaitée (bonne) à l'intérieur du système par contre dans une interface de manipulation par utilisateur elle n'est pas souhaité.

1.6.3.1 Exponential Maps

Cette représentation introduite par Grassia [36] est une extension du quaternion, dans l'Exponential Maps les rotations se formulent à l'aide d'un vecteur de trois dimensions qui décrit l'axe de rotation et dont la norme contient l'information sur l'angle de rotation.

Comme les angles d'Euler cette représentation utilise trois valeurs afin de représenter une rotation, ce qui engendre le problème de singularité, mais contrairement aux angles d'Euler on peut s'éloigner de ce problème, en ajoutant des calculs supplémentaires pour avoir toutes les propriétés mathématiques fondamentales. La seule limite vient des techniques d'interpolation linéaire qui conservent les propriétés mathématiques des exponential maps uniquement pour des interpolations entre rotations voisines (la géodésique étant alors assimilable a un segment).

1.7 Méthodes d'édition de mouvement

L'édition de mouvement sert à contrôler un mouvement naturel qui respecte les lois mécaniques afin qu'il puisse vérifier des nouvelles contraintes : squelettes différents de celui du sujet qui a effectué les mouvements, environnement 3D différent, consigne de l'utilisateur où il peut guider des parties du squelette d'une manière interactive. Nous considérons donc une méthode d'édition de mouvement, toute méthode qui possède comme entrée un mouvement initial naturel et dont le but est de modifier ce mouvement en préservant les caractéristiques importantes du mouvement initial. Beaucoup de classifications sont données. Selon [77, 78] les méthodes de génération de mouvement peuvent être classées en :

- Les méthodes qui permettent de générer un mouvement à partir d'un mouvement de référence, ce dernier étant le plus souvent acquis sur des sujets réels par du matériel d'acquisition de mouvement ou motion capture,
- Les méthodes de la cinématique qui étudient les propriétés géométriques du mouvement indépendamment des forces qui causent ce mouvement,
- Les méthodes de la dynamique, s'appuyant sur les lois de la physique, qui s'intéressent aux causes du mouvement, c'est-à-dire aux actions mécaniques mises en jeu.

Matthieu Aubry [7] a proposé de diviser les travaux existants en trois catégories, des méthodes cinématiques, physiques et des méthodes basées sur une base de mouvement. Les classifications déjà décrites ont concentrées sur le principe de fonctionnement, par contre,

il existe des classifications qui se sont concentrées sur le point de vue opérationnel, où l'édition de mouvement peut être classée selon le nombre des mouvements impliqués dans la modification. Sudarsky and House [97] ont définis les méthodes d'édition de mouvements en : méthodes unaires qui utilisent un seul mouvement, la manipulation de plusieurs mouvements est définie comme des opérations binaires. Une autre classification est donnée par Jeong et al [48] qui ont définie deux catégories de méthodes : inter-frame et intra frame. Les méthodes inter-frame ont besoin de deux ou plusieurs mouvements, et intra-frame ont besoin d'un seul mouvement.

A cause de l'absence d'une classification standard, chaque auteur a classé les méthodes selon ses besoins, selon l'objectif de son travail et selon la définition de son problème. Dans notre thèse, nous avons défini le problème majeur des méthodes d'édition de mouvement comme étant la manière de modifier un mouvement naturel tout en conservant certaines caractéristiques désirées du mouvement d'origine et de modifier d'autres caractéristiques. En se basant sur le nombre de mouvements contribué lors de la modification, nous avons classé les méthodes d'édition de mouvements en des méthodes de modification de mouvement et des méthodes de mélange de mouvement.

Selon le principe de fonctionnement, les méthodes de modification des mouvements sont divisé en des méthodes à base de traitement de signal et des méthodes à base de contrainte. Dans l'autre côté, les méthodes de mélange de mouvement sont classées en : des méthodes à base de lois statistiques et des méthodes à base d'interpolation. La figure 1.2 représente le schéma générale de la classification proposée.

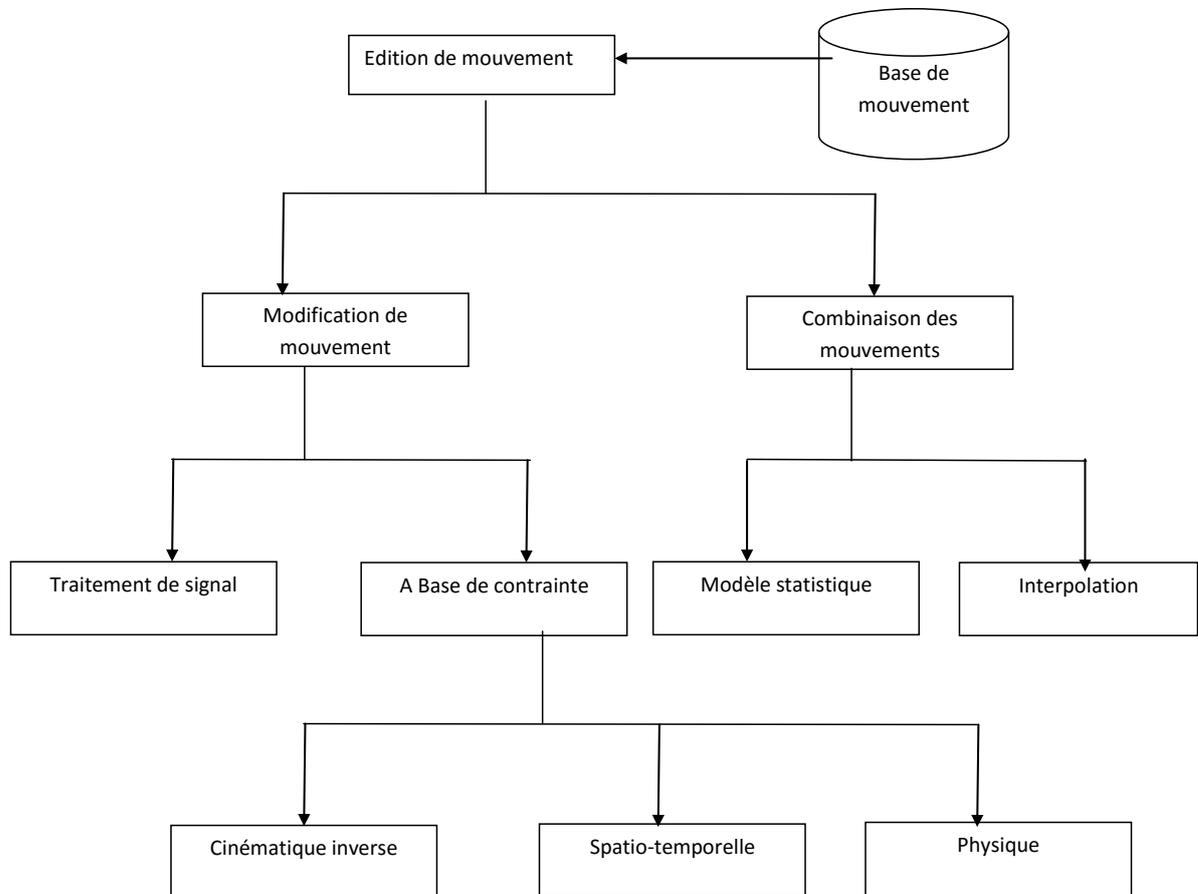


FIGURE 1.2 – Classification des méthodes d'édition de mouvement selon le principe de fonctionnement et le nombre de mouvements utilisés

1.7.1 Modification de mouvement

C'est la classe qui représente les méthodes servant à modifier un seul mouvement en entrée. Cette classe regroupe les méthodes à base de traitement de signal et les méthodes à base de contraintes.

1.7.1.1 Méthodes à base de traitement du signal

Le principe de cette classe des méthode consiste à traiter les paramètres de mouvements comme un signal échantillonné au cours du temps, cette approche propose d'utiliser des techniques à partir du domaine de traitement d'image et de signal qui sont adaptés afin

de modifier et de mélanger les mouvements des personnage virtuel. Bruderlin et Williams [19] ont introduit des filtres multi-résolutions et des interpolations multi ciblés pour traiter le mouvement. Ils se basent sur l'intuition qui stipule que les basses fréquences d'un mouvement représentent les caractéristiques principales de mouvement alors que les hautes fréquences représentent les détails du mouvement. Afin d'isoler les basses fréquences de chaque articulation pour chaque mouvement, ils ont utilisé des filtres multi-résolution, ensuite ils appliquent une interpolation multi-ciblé pour transiter d'un mouvement à un autre figure 1.3.

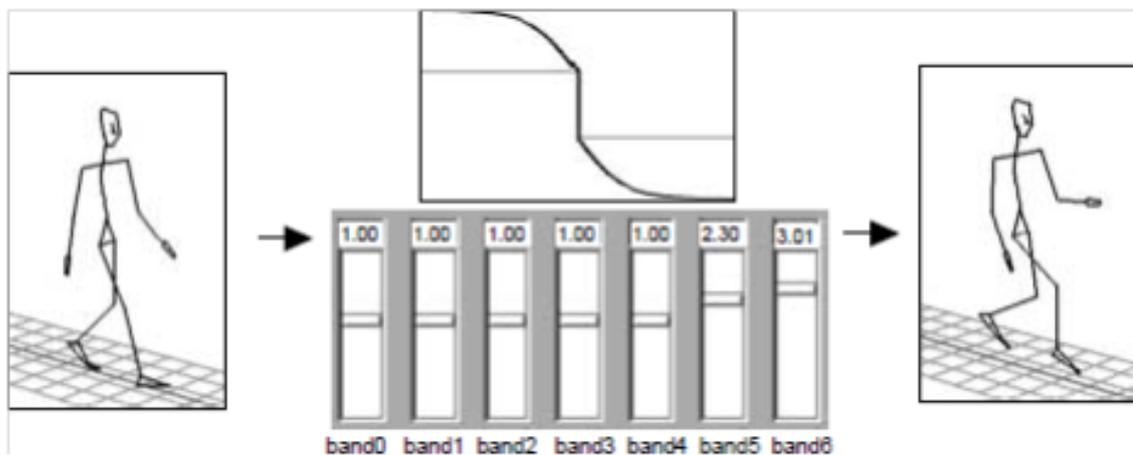


FIGURE 1.3 – Modification des basse fréquences Bruderlin et Williams [19]

En même temps et pour permettre de modifier un mouvement d'origine afin d'obtenir un nouveau mouvement adapté à la situation, Witkin et Popovic [112] ont proposé une méthode appelée déformation du mouvement *motion warping*. Afin d'effectuer cette adaptation, ils ont spécifié une série de contrainte cinématique 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. Le nouveau mouvement est alors constitué de ces postures-clés.

Un autre outil de traitement de signal est utilisé dans ce domaine, c'est les séries de Fourier. Unuma et coll. [102, 101] ont appliqué les séries de Fourier sur les trajectoires angulaires d'un personnage en train de faire une activité de locomotion afin d'extraire certaines caractéristiques. Ces dernières sont utilisées par la suite afin de générer des nouveaux mouvements en interpolant ou en extrapolant avec d'autres mouvements dans le domaine fréquentiel. Par exemple, ils ont modifié une marche normale en marche fatiguée.

(Figure1.4).



FIGURE 1.4 – Course avec émotions [101]

Amaya et al. [3] ont présenté une technique d'extraction de l'aspect émotionnel d'un mouvement et l'ont appliqué à un autre mouvement pour ajouter des émotions. Leur méthode est divisé en trois étapes. Premièrement, ils ont capturé le même mouvement avec différentes émotions : neutre, triste, en colère, etc. Ensuite, l'aspect émotionnel est extrait par la soustraction des mouvements neutres et émotionnels. Enfin, ce composant est appliqué à un autre mouvement neutre afin d'ajouter des aspects émotionnels similaires.

Ces méthodes sont généralement difficiles à utiliser parce que la manipulation du signal variant dans le temps est délicate et non-intuitive lorsque l'animateur veut modifier un mouvement avec des requêtes précises. En plus, ils n'assurent pas que les contraintes cinématiques du mouvement d'origine soient conservées, par exemple lorsqu'on utilise une déformation du mouvement, si les postures clés ne sont pas placées correctement, le mouvement final peut violer des contraintes géométriques importantes comme le glissement ou la pénétration des pieds dans le sol.

1.7.1.2 Méthodes à base des contraintes

Les méthodes d'édition de mouvement à base de contraintes décrivent explicitement les caractéristiques de mouvement à changer ou à préserver. Contrairement aux méthodes qui ne se basent pas sur les contraintes et qui ne rendent pas de telles caractéristiques explicites, par exemple, dans les opérations de filtrage ou mélange de mouvements appliqués à un mouvement, elles modifient ce dernier, mais sans décrire les opérations en terme de caractéristiques, seulement en terme de paramètres permettant de représenter le mouvement.

Les méthodes à base de contraintes se focalisent généralement sur les contraintes géo-

métriques ou spatiales. Les contraintes spatiales fournissent les configurations des postures dans un temps donné. Par exemple, on spécifie que la main doit être dans une position donnée dans un temps défini. D'ailleurs, il existe d'autres types de contraintes tel que les contraintes physiques utilisé afin de respecter les lois de la physique.

Les méthodes d'édition de mouvement à base des contraintes peuvent être classées selon le type des contraintes spécifiées en : cinématique inverse, méthodes spatio-temporelle et lois physique. Avant de citer les méthodes à base de contraintes, nous allons définir une étape très essentielle : *détection des contraintes*.

Détection des contraintes

La première étape dans toutes les méthodes à base de contraintes consiste à déterminer les informations spatiales et temporelles concernant les contraintes dans le mouvement d'origine. Dans le cas général, ces données sont définies manuellement par l'animateur [87, 80].

Dans le cas de l'existence d'une description numérique de l'environnement, une méthode simple de détection des contraintes peut calculer la distance entre les objets dans l'environnement et le modèle articulé. Cependant, faire ce calcul à chaque frame peut être inefficace.

Une application classique de la détection des contraintes consiste en l'identification des *footplant* dans les mouvements, Kovar et utilisent des filtres spécifiques sur la position, et la vitesse des pieds afin de détecter ces contraintes [60, 73]. De la même façon, Lee et al [66] ont étendus le travail de Kovar aux segments des personnages et des objets de l'environnement et non pas seulement sur la détection de *footplant*. Ils prennent en considération leur vitesse et leur position afin de décider si les segments sont en contact avec un objet de la scène ou non. Cependant, les filtres utilisés dans ces techniques sont liés à la nature des mouvements. Par conséquence, ils doivent être finement régler en fonction de la nature du mouvement du personnage : marcher, courir, sauter..etc.

Dans le travail de Bindiganavale et al [14] les auteurs ont présenté une méthode qui adapte le mouvement d'un sujet à un autre sujet qui n'a pas la même morphologie où la différence réside au niveau des proportions des membres. Ils se sont focalisés sur des mouvements qui contiennent des interactions avec les objets de l'environnement. Le point fort dans leur technique est le fait d'identifier les collisions entre les extrémités des membres et les objets de l'environnement. Cependant, pour éviter les vérifications à chaque frame, ils supposent que les frames d'intérêt potentiel sont situés dans le passage à zéro de la

dérivée seconde des extrémités. Finalement, ils ont utilisé des objets marqués qui ont des sites d'intérêts prédéfinies, afin d'éviter les vérifications des collisions avec tous les objets de la scène. Lecallennec et Boulic [65] ont développé une méthode pour la détection automatique des contraintes en utilisant uniquement la différence dans les transformations entre deux frames. Ils ont développé aussi un filtre de prédiction qui permet d'augmenter la robustesse de la phase de la détection.

Après la description de l'étape de détection des contraintes, nous présenterons les différentes méthodes d'édition de mouvement à base de contrainte.

Cinématique inverse

La cinématique inverse est l'application clé dans les systèmes d'édition des mouvements à base des contraintes, elle est utilisée afin de résoudre les contraintes géométriques. La cinématique inverse implique la résolution des contraintes géométriques dans un temps défini.

Toutes les méthodes d'édition de mouvement à base de contraintes doivent contenir un solveur du problème de la cinématique inverse pour résoudre les contraintes sur les positions des extrémités. La nature de la cinématique inverse mène à des défis que n'importe quel solveur doit prendre en considération. Un état de l'art plus profond sur cette classe est donné dans le prochain chapitre.

Dans la définition classique de la méthode de la cinématique inverse, celle-ci ne prend pas en considération les relations entre les frames. Une façon d'étendre l'information à un autre frame, c'est l'utilisation de filtrage afin de créer un mouvement continu avec des contraintes renforcées. Ce groupe de techniques est appelé « perframe cinématique inverse +filtrage » (PFIK+F). 1.5

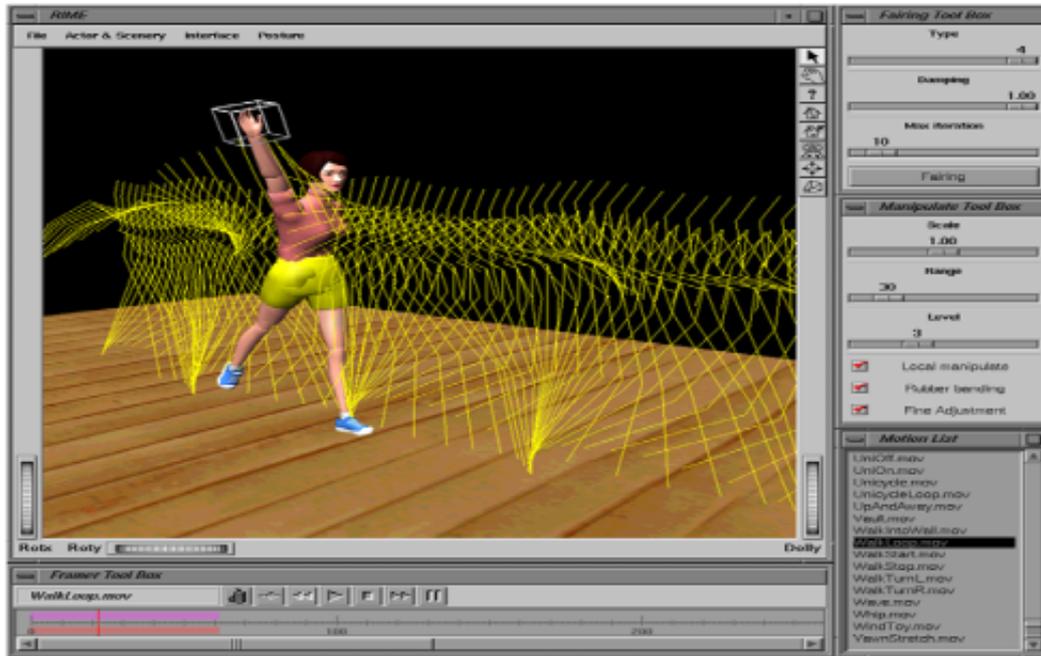


FIGURE 1.5 – Système de modification de mouvement à base de contraintes [67]

Choi et al [24] utilisent une méthode de la cinématique inverse afin de renforcer la configuration de l'extrémité dans un intervalle de temps bien défini, ensuite, ils utilisent un filtre entre les poses au début et à la fin de l'intervalle, afin d'assurer la continuité de mouvement final. Beaucoup d'autres méthodes utilisent des approches similaires et utilisent des méthodes à partir du domaine de traitement de signal [34, 61, 15]. Des approches différentes de filtrage impliquent d'ajuster une spline aux données afin de créer des mouvements continus[112, 67]. Le travail de Shin et al [93] propose de résoudre le problème de la cinématique inverse en imposant un compromis entre le temps de calcul et le réalisme. Ils associent une valeur d'importance au poignets et chevilles du personnage virtuel selon leur proximité avec des objets de l'environnement. Ils ont proposé un algorithme afin de trouver la position de la racine du personnage, en essayant de respecter les contraintes placées sur les extrémités considérant leur importance. Figure 1.6

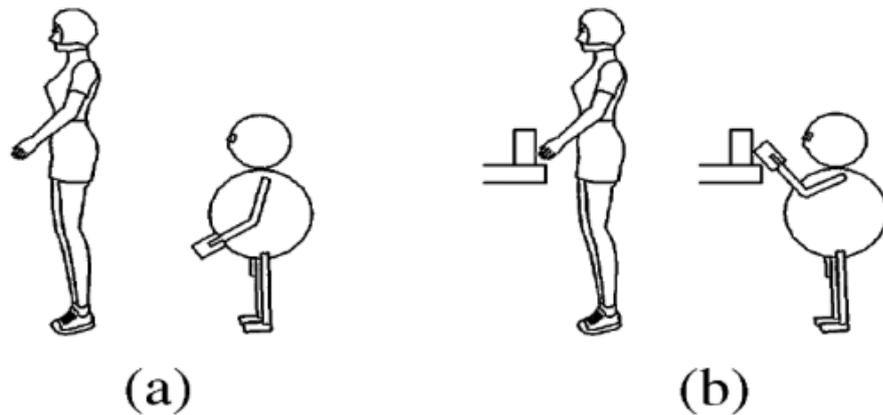


FIGURE 1.6 – Importance associée aux extrémités en fonction de leur proximité avec un objet [93]. (a) Les orientations des bras de la femme sont directement appliquées sur le personnage dû à l'absence d'un objet à côté de ces extrémités. (b) La conservation de la position des poignets est réalisée puisqu'ils se trouvent à proximité d'un objet.

Monzani et al [75] ont utilisé un squelette intermédiaire afin d'adapter un mouvement à un autre personnage, la différence entre les deux personnages réside dans la géométrie et la topologie. La première étape sert à spécifier les correspondances entre les articulations du personnage d'origine (Performer Skeleton) et celle du squelette final (end user skeleton). La figure 1.7 montre cette étape pour les articulations d'un bras. Dans la deuxième étape, les positions sont mises en correspondance afin de transposer les orientations du mouvement initial sur le nouveau personnage (figure 1.7). Dans la dernière étape, ils spécifient des contraintes cinématiques afin de conserver certaines caractéristiques du mouvement initial. La résolution des contraintes est faite par l'utilisation de la cinématique inverse. Afin d'assurer la transition continue entre les trajectoires des extrémités du personnage original et les trajectoires des contraintes, ils ont ajouté des périodes "ease-in(eas out) " avant (respectivement après) que les contraintes soient activées (respectivement désactivées). Durant les périodes "ease-in" les trajectoires des extrémités sont interpolés linéairement afin d'atteindre les objectifs spécifiques d'une manière continue. Durant les périodes "ease-out", ils ont interpolés linéairement les postures au lieu d'interpoler les trajectoires. Selon les auteurs, cette méthode donne de bon résultats seulement quand le mouvement adapté est proche du mouvement original. Par contre, l'interpolation linéaire peut produire des grandes discontinuités parce qu'elle ne prend pas en compte la vitesse des extrémités.

1.7. MÉTHODES D'ÉDITION DE MOUVEMENT

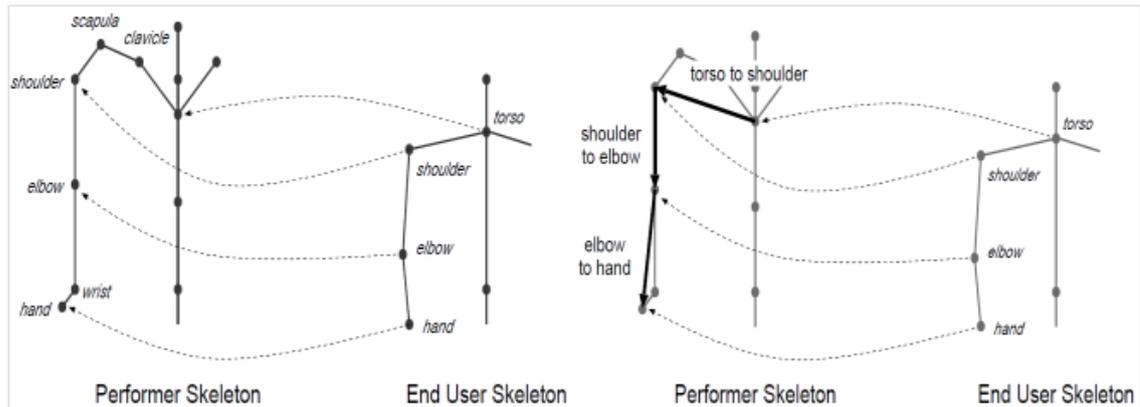


FIGURE 1.7 – Adaptation des angles de jointures d'un bras en utilisant un squelette intermédiaire [75]

Lee et Shin [67] proposent de décomposer les trajectoires articulaires en splines hiérarchiques (figure 1.8). Leur méthode permet de modifier localement les mouvements et de propager la déformation sur les postures voisines de manière continue. La décomposition hiérarchique permet en outre de travailler localement ou globalement suivant le niveau choisi (modifications des basses ou hautes fréquences). La résolution des contraintes est effectuée à l'aide de la cinématique inverse combinée avec des résolutions analytiques pour les membres.

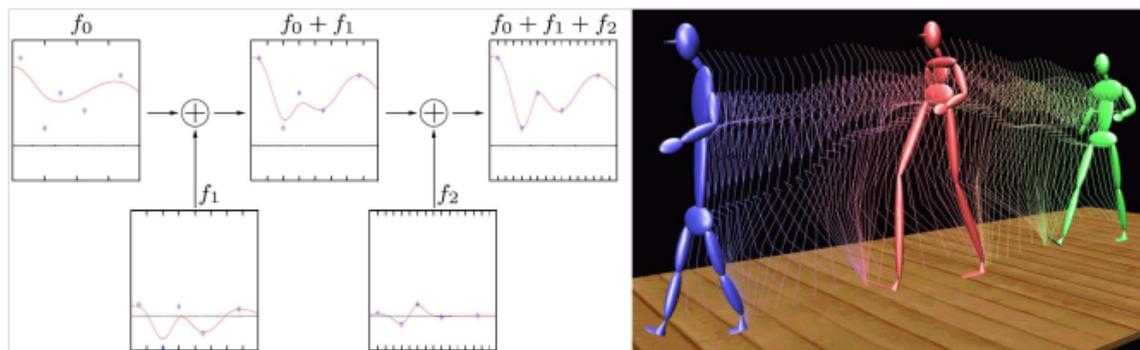


FIGURE 1.8 – Décomposition des trajectoires articulaires en splines hiérarchiques [67]

Choi et ko [24] présentent le premier travail sur l'adaptation de mouvement à une autre morphologie en ligne *online motion retargeting*, pour cela ils ont proposé une méthode de la cinématique inverse en se basant sur le contrôle de vitesse de mouvement. Un autre travail dans cette classe de méthodes, c'est le travail de Kulpa [62] qui a proposé une adaptation de mouvement, grâce à une implémentation de l'algorithme de CCD (cyclic coordiante descent) afin d'adapter un mouvement en temps réel. Cette méthode n'utilise aucune

méthode de filtrage, elle donne de bon résultats pour l'adaptation d'un mouvement à une autre morphologie en temps réel. Afin de faciliter l'adaptation, ils ont divisé le squelette en groupes, chacun est vu comme une chaîne cinématique indépendante. Cette subdivision a l'avantage de faciliter les calculs, mais comme il n'y a pas une synergie entre les groupes, elle peut donner des résultats irréalistes. Lieu [71] a proposé une technique d'édition de mouvement par évitement de collision afin de prévenir l'interpénétration des membres du corps du personnage durant l'édition. Afin de garantir la continuité des mouvements finaux et la similarité entre le mouvement édité et le mouvement d'origine, les techniques des cartes de déplacement et les filtres de Kalman sont utilisés. Le Callenc et Boulic [64] ont proposé une technique PFIK+F d'édition de mouvement interactive avec des contraintes prioritaires. Les caractéristiques du mouvement original sont préservées en ajoutant la différence entre le mouvement avant et après l'édition comme une contrainte avec la plus faible priorité en utilisant un vecteur d'optimisation, ensuite ils ont utilisé un filtre spline afin d'imposer la continuité de mouvement. En définissant des contraintes avec des différentes priorités, il est possible de générer des mouvements plus réalistes en laissant la main à l'utilisateur pour spécifier la priorité de chaque contrainte dans le cas de conflits.

Le travail de Cavalho [22] diffère du travail précédent dans l'utilisation d'une base de mouvement afin de synthétiser un mouvement. Il travaille dans un espace appelé latent (obtenus par méthodes de type analyse en composantes principales ACP) où il a proposé une méthode de la cinématique inverse à base de priorité dans l'espace latent LPIK. Par contre son travail souffre du problème lié à l'utilisation des bases de mouvement, ils ont besoin d'une base de mouvement large, et d'une étape de pré traitement et d'apprentissage.

Méthodes spatio-temporelles (Space time constraint)

La méthode spatio-temporelle peut être utilisée pour synthétiser de nouveaux mouvements ou pour éditer des mouvements déjà existants, en utilisant des contraintes dynamiques ou cinématiques. En plus, la formulation des contraintes spatio-temporelles est à l'origine de beaucoup de travaux dans l'animation des humains virtuels.

Cette classe regroupe les méthodes qui traitent le mouvement comme un seul bloc contrairement aux autres méthodes qui calculent le nouveau mouvement frame par frame.

witkin [111] a proposé de déformer un mouvement de référence en utilisant une technique d'optimisation visant à trouver le meilleur mouvement satisfaisant un ensemble défini de contraintes. L'optimisation des contraintes spatio-temporelles offre une méthode

générale pour déformer un mouvement d'origine.

Pour minimiser le coût de calcul, Cohen [26] a 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. Gleicher et Litwinowicz [35] ont proposé d'appliquer des contraintes spatio-temporelles pour modifier des mouvements préexistants. L'entrée de leur système est un mouvement qui existe déjà, ce qui rend le problème de la création du mouvement plus simple, car ils n'ont pas besoin de spécifier les détails du mouvement à partir de zéro, mais plutôt devraient s'assurer que le mouvement final reste plus proche de l'initial. Gleicher [35] a appliqué la méthode des contraintes spatio-temporelles pour adapter un mouvement à une autre morphologie. L'utilisateur définit un ensemble de contraintes que le mouvement initial doit préserver. En se basant sur cet ensemble de contraintes, le solveur spatio-temporelle essaye de trouver des mouvements qui satisfont l'ensemble des contraintes précédentes en minimisant une fonction objective. Dans cet algorithme Gleicher a utilisé une fonction d'objective permettant de minimiser la consommation d'énergie des articulations par la différence entre le mouvement d'origine et final.

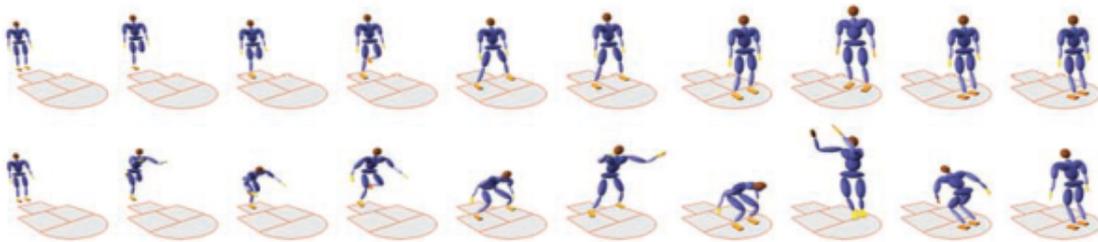


FIGURE 1.9 – Génération de mouvements naturels à base des contraintes spatio-temporelle dynamiques.[69]

Liu and Popovic [69] figure 1.9 ont utilisé un modèle dynamique simplifié et des contraintes spatio-temporelles pour générer des mouvements naturels à partir des séquences de mouvement simples définis par un animateur (keyframe). Ces séquences de mouvement sont, dans un premier temps, analysées afin d'extraire les contraintes de positionnement et de non glissement pour séparer les séquences de mouvement en des étapes avec des contraintes et sans contraintes, ensuite, ils génèrent des transitions entre les étapes en suggérant à l'utilisateur un ensemble de postures de transition déjà apprises. Finalement, ils calculent le mouvement final en minimisant déplacements des masses, la vitesse des DDL et en assurant l'équilibre statique. Contrairement à [69], Abe et al [1] ont généré un ensemble de mouvements en se basant sur un seul mouvement en entrée ;

ensuite ils ont générer un ensemble varié des mouvements en temps réel en utilisant des interpolations simples.

Les travaux accomplis dans ces papiers donnent de bons résultats, mais possèdent des limites. Le premier problème concerne le temps de calcul qui est très grands, ce qu'empêche leur utilisation pour des longues séquences de mouvement. En outre, lorsque le nombre de DDL du personnage augmente, le temps d'exécution devient rapidement prohibitif. En plus, en utilisant les méthodes spatio-temporelles le personnage virtuel doit avoir une connaissance complète sur le futur afin d'optimiser son action. Une autre limite concerne la difficulté de contrôler les calculs en raison de la nature des contraintes qui est hautement non-linéaire. Les méthodes spatio- temporelles ont besoin de résoudre un seul problème mathématique pour tout le mouvement, ce qui conduit à un problème d'optimisation très complexe et généralement très difficile à résoudre.

Contraintes physiques

Généralement, on utilise les lois physiques pour démontrer le réalisme du mouvement. Quelques méthodes imposent des contraintes sur le centre ZMP "zero moment point" du personnage virtuel pour rester dans le support du polygone qui est un critère. Afin de corriger l'aspect physique d'un mouvement, des chercheurs ont proposé une prise en compte du ZMP sous forme de contraintes spatio-temporelles. La résolution du système de contraintes permet d'éditer la trajectoire du ZMP et d'assurer l'équilibre dynamique lors d'un mouvement quelconque [98] qui est une condition de l'exactitude physique. Chin et al [92] ont estimé la distribution des masses du personnage en utilisant des mouvements de référence connus comme physiquement justes. Ensuite, ils ont modifié le mouvement en utilisant des lois physiques. Pendant la phase aérienne ; ils ont assuré que son impulsion angulaire est conservé. Pendant l'appui, ils ont utilisé des formules simples des ZMP pour assurer que les personnages sont dynamiquement équilibrés. Dans [115] les auteurs utilisent un double filtrage des paramètres pour vérifier les contraintes dynamiques. Le premier filtre prédit le prochain vecteur d'état en fonction des contraintes cinématiques et dynamiques telles que le respect du ZMP et de la position initiale ainsi que les limites des moments angulaires. Le second filtre corrige les contradictions résultantes entre les positions, les vitesses et les accélérations..

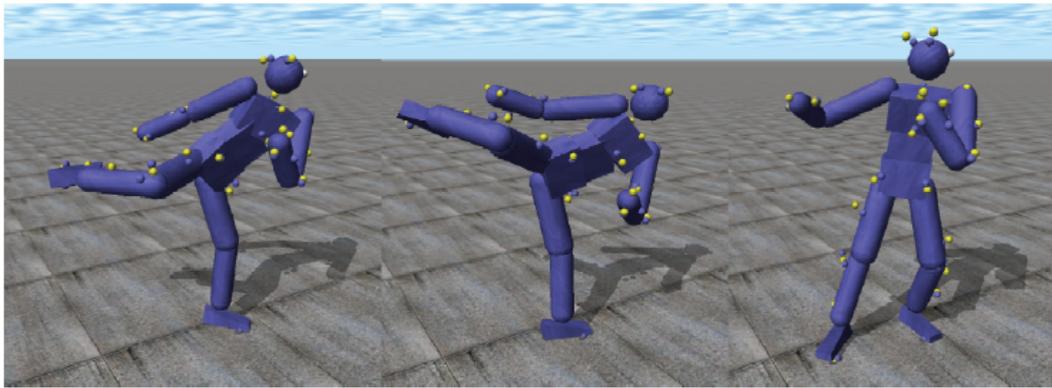


FIGURE 1.10 – Modification de mouvement à base des contraintes physiques [119]

Zordan et al [117] ont contrôlé la simulation physique en utilisant une base de mouvement déjà capturés. Le personnage virtuel est capable de réagir d'une manière dynamique avec les forces extérieures comme le poinçon. Zordan and Van Der Horst [119] ont proposé une nouvelle approche afin de convertir les positions des marqueurs 3D aux angles d'articulations en utilisant la dynamique. La méthode utilise une approche de la dynamique inverse (comme IK) pour calculer les postures de chaque frame. Chaque marqueur est contrôlé par une force masse ressort (spring force) qu'elle l'attire à une position précise figure 1.10. En utilisant des forces internes, une certaine cohérence est ajoutée pour éviter la violation des limites des articulations. En plus, ils ont appliqué footplants en utilisant des forces extérieures de frottement. Zordan et al. [118] ont présenté une technique hybride qui utilise la simulation physique et des mouvements déjà capturés pour simuler des effets inattendus dans l'humain virtuel. Lorsqu'un effet survient, la simulation physique conduit l'animation tant qu'une routine spéciale de recherche détermine le meilleur retour plausible à la base de données de mouvement suite de l'effet.

Shapiro et al [91] ont focalisé dans la plateforme "DANCE" sur le développement des contrôleurs physiques pour les personnages virtuels. L'objectif essentiel de cette plateforme est d'apprendre aux utilisateurs sans expérience de développer des contrôleurs dynamiques. Sok et al [96] ont présenté un framework pour l'édition interactive des moment d'inertie et les forces extérieures dans une séquence de capture de mouvement. Ils utilisent dans leur système une optimisation de trajectoire avec un éditeur de contraintes. Kenwright [55] a proposé la modélisation des mouvements du squelette 3D en temps réel avec le respect des propriétés d'équilibre.

1.7.2 Combinaison de mouvements

Les méthodes regroupées dans cette classe, utilisent un ensemble de mouvements afin de synthétiser un autre mouvement. Ces méthodes peuvent être à base de lois statistiques ou à base d'interpolation.

1.7.2.1 Synthèse à base des lois statistiques

Ces approches permettent d'extraire une caractéristique d'un ou de plusieurs mouvements qui seront réutilisés par la suite pour créer un autre mouvement. Cette caractéristique est couramment désignée sous le nom de style du mouvement. Il existe différentes manières de modéliser le style, il peut être sous la forme de gains [57], des contraintes articulaires [83] ou d'un espace latent [68, 37, 114, 100]. Principalement, ces approches cherchent à modéliser l'espace des postures (contraintes) ou les synergies du mouvement (gains et espaces latents).

L'approche de Ong [83] propose de prendre une base de mouvements, puis à partir de toutes les postures prises par le personnage, et d'établir un apprentissage statistique de la configuration articulaire la plus probable en fonction de la position de l'extrémité. Ainsi lors de la boucle d'optimisation, la cinématique inverse peut être guidée vers la posture la plus probable. Min et al [74] ont introduit un modèle à base de statistiques afin de modifier des mouvements, en donnant la trajectoire d'une partie spécifique du personnage virtuel. Des mouvements d'exemples sont mélangés afin de synthétiser un nouveau mouvement.

Cependant ce type d'approche pose des problèmes. Tout d'abord, les informations extraites à partir de l'ensemble des mouvements sont destinées uniquement à un contrôle spécifique et on ne peut pas les utiliser dans d'autres contrôleurs, ensuite la quantité des mouvements pour un apprentissage fiable doit être assez grande et enfin le temps de calcul pour générer un modèle utilisant les lois statistiques est très élevé.

1.7.2.2 Synthèse de mouvement à base d'interpolation

Un mécanisme d'interpolation permet d'obtenir le mouvement adéquat pour résoudre une nouvelle tâche. Cette solution sert à multiplier le nombre d'exemples. Au lieu de déformer un seul mouvement ; il convient d'en mélanger plusieurs. Des fois on peut utiliser le même mécanisme afin d'extrapoler les données en dehors de l'espace des exemples. Dans ce cas, le réalisme de mouvement est plus souvent compromis. Les travaux [112, 19] représentent les premiers travaux dans cette classe, qui sont des méthodes classiques de

traitement de signal mais ils permettent aussi de combiner deux mouvements. D'autres approches permettent d'obtenir des résultats sans traitements intermédiaires comme les approches basées sur les plus proches voisins [2, 60] ou à base des fonction radiales [86, 88].

Une autre façon d'interpoler des mouvements est le motion graph ou les graphes des mouvements où on peut synthétiser l'ensemble des postures dans un même graphe [60]. Les nœuds de graphe sont les différentes postures alors que les arcs représentent les possibilités de passage d'une posture à une autre. Afin de définir le mouvement du personnage virtuel, on parcourt le graphe à la recherche des postures lui permettant d'atteindre le but fixe.

Le travail de Kovar [60] permet de synthétiser uniquement différents mouvements de la marche (changement de trajectoire), Safanova [89] a proposé une extension de ce dernier travail, en proposant une méthode capable de synthétiser un mouvement comprenant plusieurs tâches et permettant de résoudre des contraintes imposées par l'utilisateur. Cependant, cette approche souffre d'une complexité très élevée, ce qui ne permet pas de synthétiser un mouvement en temps réel.

Les méthode à base d'interpolation donnent des résultats réalistes. Cependant, il reste l'inconvénient majeurs lié au besoin d'une très large base d'exemples de la même tâche à réaliser, ce qui augmente la complexité de chercher la bonne posture.

1.8 Bilan

Avant d'entamer les différentes méthodes de modification de mouvement, nous avons présenté les différents types de représentations des personnages virtuels, une représentation squelettique qui représente le personnage comme un arbre de jointures caractérisées par la position et l'orientation, la représentation physique dont on ajoute la définition des caractéristiques physiques comme les masses, les moments inertiels...

Concernant la représentation des angles de rotations, nous avons constaté qu'il existe plusieurs représentation chacune possède ses propres avantages et inconvénients :

Les angles d'Euler présentent une représentation intuitive des degrés de liberté mais posent un problème de singularité, le même problème avec la représentation matricielle qui est coûteuses en multiplication aussi, cependant dans la représentation par quaternion et les exponential maps le problème de singularité ne figure pas.

Chaque représentation peut être ainsi plébiscité selon le contexte de l'utilisation. La possibilité de passer d'une représentation à une autre donne la possibilité d'utiliser plusieurs représentations dans un système d'animation.

Pour les différentes techniques permettant de modifier/éditer un mouvement, nous avons proposé une classification de ces techniques selon le principe de fonctionnement. Il existe des méthodes qui représentent le mouvement sous la forme d'un signal, en modifiant ce signal on obtient un nouveau mouvement. D'autres méthodes sont basées sur les contraintes, où l'utilisateur doit définir explicitement quelques contraintes à préserver ou à respecter. Les méthodes spatio-temporelles sont présentées aussi, elles donnent de bons résultats parce qu'elles traitent le mouvement comme étant un seul bloc mais ont un temps de calcul très élevé et ils ont besoin de savoir par anticipation tout les information sur la posture à modifier ce qui n'est pas approprié pour les applications interactives. Enfin, des méthodes à base d'interpolation et à base de lois statistiques sont présentées, ces méthodes donnent des résultats naturels, mais nécessitent une base de mouvement large et une étape longue de pré-traitement, un autre inconvénient de ces méthodes est que le modèle construit est appliqué sur un seul mouvement présenté dans la base, pour modifier un autre mouvement, on doit construire un autre modèle.

1.9 Conclusion

Notre objectif est de modifier un mouvement d'un personnage virtuel. Dans ce chapitre nous avons présenté un état de l'art sur la modification des mouvements. Cet état de l'art nous a permis, dans un premier temps, de faire le point sur quelques définitions et les différentes représentations des angles de rotation qui existent dans la littérature. Une fois ce constat établi, nous sommes passé aux techniques de la modification qui sont le cœur de notre travail.

Les méthodes de modification du mouvement à base des contraintes servent à ajouter au mouvement des contraintes à respecter dans quelques frames. Plusieurs méthodes de modification de mouvement à base des contraintes sont présentées. Les méthodes à base de la cinématique inverse sont les plus utilisées afin de modifier un mouvement d'une manière interactive, c'est pourquoi nous allons nous concentrer sur l'état de l'art des méthodes de la cinématique inverse.

Dans le prochain chapitre, Une description détaillé de l'aspect de cinématique d'un mouvement et ensuite nous décrirons les différentes classes de la cinématique inverse ainsi que les travaux les plus importants dans chaque classe.

Cinématique Inverse

2.1 Introduction

Le besoin de manipuler des structures articulaires d'une manière complexe est très demandé dans l'industrie pour contrôler des chaînes robotiques ou dans les domaines de la simulation pour contrôler des structures articulaires de personnages virtuels. Dans le cas des personnages virtuels, les structures articulaires représentent un ensemble de jointures flexibles reliées entre elles par des segments. Une chaîne articulaire peut donner une infinité de postures, où chacune est représentée par la position de la racine et les rotations de toutes les jointures de la racine à l'extrémité.

La cinématique inverse (IK) est l'une des techniques utilisées afin de générer une nouvelle posture à partir de la connaissance de la position finale de l'extrémité. L'objectif d'IK est de calculer les angles formulés par les articulations entre elles, afin de déplacer l'extrémité de la chaîne articulaire aussi près de la position souhaitée.

Dans ce chapitre, nous allons donner un état de l'art sur les méthodes existantes pour résoudre le problème de la cinématique inverse leurs avantages et leurs inconvénients.

2.2 Vocabulaires de la cinématique

Avant d'entamer les détails de la cinématique inverse nous devons premièrement définir ce que veut dire une chaîne cinématique et degré de liberté qui sont beaucoup utilisés dans ce chapitre et dans la thèse.

2.2.1 Chaîne cinématique

La cinématique est définie comme l'étude des mouvements : c'est une branche de la physique qui traite le mouvement d'un système sans faire référence aux forces ou aux masses. Les modèles cinématiques sont construits avec des segments et des jointures dans une structure hiérarchique en utilisant des relations père/fils. Les segments sont considérés comme des membres rigides et les jointures sont les liens entre les segments. En tournant une jointure autour d'un axe arbitraire tous ses fils sont aussi tournés autour du même axe car leur relation est définie de cette façon puisque qu'elle hérite de toutes les transformations de ses parents. La figure 2.1 représente une chaîne cinématique simple avec quatre jointures dont la première représente la racine et la dernière représente l'extrémité de la chaîne.

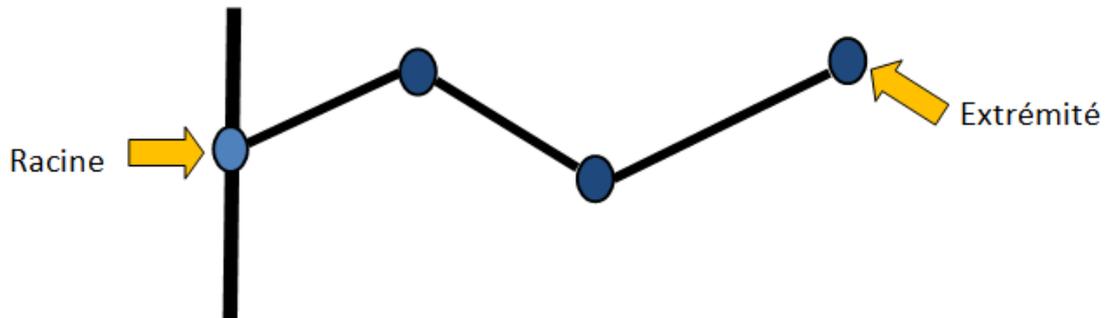


FIGURE 2.1 – Chaîne cinématique simple avec une racine et une seule extrémité

Les jointures ont la possibilité d'être orientées ou translattées, on dit qu'une jointure à un degré de liberté si elle peut être orientée autour d'un axe ou déplacée le long d'un axe.

2.2.2 Degré de liberté

Le degré de liberté, en abréviation DDL, représente le nombre de paramètres indépendants qui détermine une configuration d'une jointure, cette dernière peut avoir des degrés de liberté rotationnelles et de translation. Lorsqu'on applique une rotation ou une translation dans l'espace à trois dimensions, on définit le degré de liberté comme le nombre d'axes autour duquel on peut appliquer une rotation, plus le nombre d'axes le long desquels on peut faire une translation. Cela veut dire que :

- une jointure avec 1-DDL rotationnelle, qui peut être appelée aussi « hinge joint », peut tourner autour d'un seul axe.

2.3. CINÉMATIQUE DIRECTE

- une jointure avec 1-DDL de translation, qui peut être appelée « prismatic joint » ou slider joint », peut déplacer le long d'un seul axe.
- De même, il existe une jointure avec 2-DDL rotationnelle, aussi appelée "revolute joint", qui peut tourner autour de 2 axes différentes.
- une jointure avec 2-DDL de translation, qui peut être appelée "planar joint", peut se déplacer le long de deux différents axes.
- une jointure 2-DDL de translation et de rotation, qui est appelée "jointure cylindrique", qui peut être tournée autour d'un seul axe et faire une translation le long d'un seul axe.
- En plus, on peut avoir une jointure avec 3 DDL rotationnelle, qui peut être appelé « ball-and-socket joint », qui peut tourner autour de trois axes.
- On peut avoir une jointure avec 3DDL de translation qui peut faire une translation le long de 3 axes.
- En outre, on peut avoir une jointure avec 6-DDL qui peut tourner autour des 3 axes et faire une translation le long de 3 axes.

La figure 2.2 représente les principaux types des jointure.

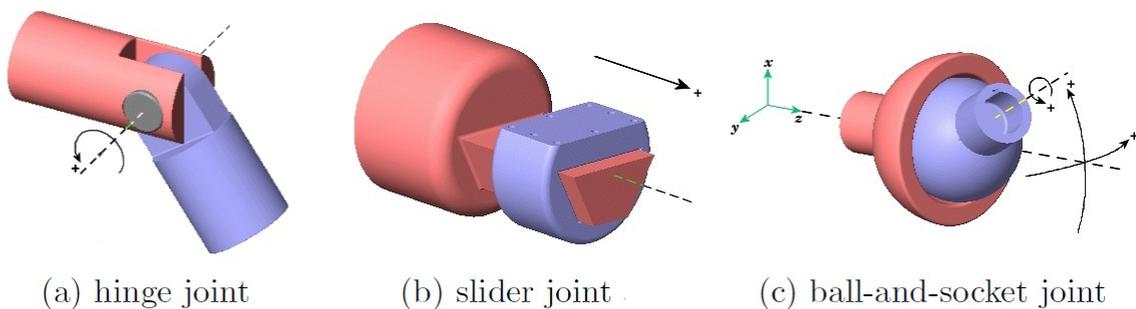


FIGURE 2.2 – Types de jointure

Le corps de personnage peut avoir des degrés de liberté rotationnel pour toutes les jointures. Cela veut dire, qu'on peut avoir 1-DDL, 2-DDL ou 3-DDL jointures rotationnelles.

2.3 Cinématique directe

Après la modélisation du squelette du personnage, nous pouvons l'animer, le contrôler avec plusieurs méthodes d'animation. Récemment, la capture de mouvement est devenue une méthode intéressante pour animer les personnages virtuels, en effet, un de ses grands

2.3. CINÉMATIQUE DIRECTE

avantages est la rapidité de produire des mouvements naturels. Cependant, le processus menant de l'acquisition des données vers l'adaptation de mouvement, introduit généralement des artefacts que les animateurs doivent corriger. Par conséquent, les animateurs ont besoin encore d'un contrôle important des résultats afin de corriger et ajuster n'importe quelle posture selon les besoins de l'animateur, tel que le problème du glissement des pieds qui se pose généralement si le sujet du mouvement capturé et le personnage virtuel n'ont pas les mêmes tailles des segments, de tels artefacts doivent être corrigés.

Une façon de corriger la posture d'un personnage virtuel consiste à utiliser la cinématique directe qui donne une posture à un squelette en modifiant uniquement les paramètres de ses articulations. Pour placer la main d'un personnage dans une position voulue en utilisant la cinématique directe, nous devons modifier les articulations de l'épaule, du coude puis du poignet afin de les placer dans des angles permettant d'obtenir la pose voulue. En d'autres termes, l'animateur spécifie le vecteur de configuration -posture- ($\theta = pos_{,acine}, \theta_1, \dots, \theta_n$) afin d'atteindre son objectif (figure 2.3). L'équation de la cinématique directe est définie par

$$pos_e = f(\theta) \quad (2.1)$$

Où pos_e est la position actuelle de l'extrémité et f est la fonction qui calcule pos_e étant donné le vecteur de configuration θ .

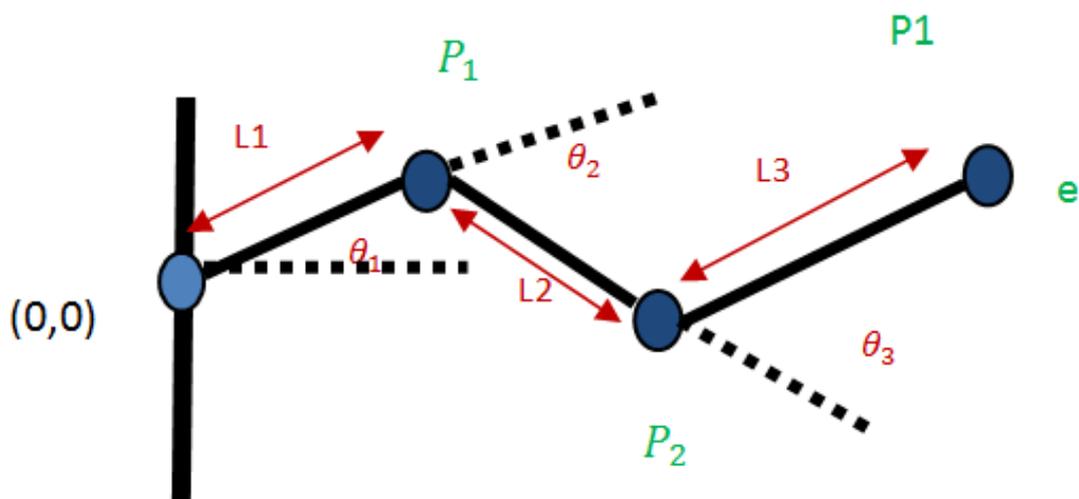


FIGURE 2.3 – Principe de la cinématique directe

Cette méthode est simple à mettre en œuvre. Le calcul de la solution est très rapide

parce que la fonction f est bien définie : c'est la composition de toutes les transformations de l'extrémité jusqu'à la racine de l'hierarchie.

Malheureusement, l'utilisation de la cinématique directe devient très complexe lorsque les chaînes hiérarchiques du squelette deviennent trop longues. En plus, l'animation créée par la cinématique directe est spécifique à un personnage dans un environnement particulier. Ainsi, toute modification de personnage ou de son environnement nécessite la réédition du mouvement. Un autre problème, la modification d'un angle d'une des jointures modifie la position de tous les éléments inférieurs dans la hiérarchie. Par exemple, si l'animateur a besoin de positionner la main afin qu'elle atteigne un objet, il aura besoin de spécifier les rotations de la racine de l'hierarchie, tronc, clavicule, épaule, coude et le poignet. Ensuite, si la main ne peut pas être positionnée correctement à cause des rotations de l'épaule qui ne sont pas adéquates, l'animateur a besoin de changer toutes les rotations des jointures de l'épaule jusqu'à la main, en effet, bouger l'épaule fait bouger aussi le coude, le poignet et la main en même temps. C'est pourquoi, l'animateur doit effectuer des ajustements ultérieurs afin d'obtenir les résultats attendus. Pour résoudre ce problème, une autre technique est disponible : la cinématique inverse qui propose de contrôler le personnage virtuel à partir de la position de ses extrémités (tête, mains, pieds). En positionnant le personnage relativement à l'environnement, les problèmes posés précédemment sont annulés.

2.4 Cinématique inverse

Un problème de cinématique inverse revient à trouver la configuration des articulations de squelette permettant de positionner une articulation selon une orientation et/ou une translation définie. Pour cela, on a besoin de résoudre l'équation :

$$\theta = f^{-1}(pos_e) \quad (2.2)$$

Où pos_e est la position désirée de l'extrémité (tâche) et f^{-1} est la fonction qui calcule le vecteur de configuration θ compte tenu de pos_e . Dans la figure 2.4 nous devons calculer $\theta_1, \theta_2, \theta_3$ dont le but que la position de l'extrémité e sera à la position souhaitée.

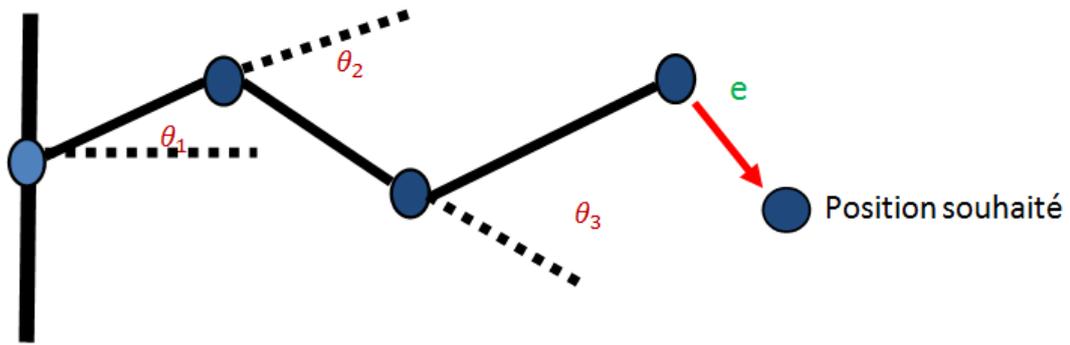


FIGURE 2.4 – Cinématique inverse

La cinématique inverse est très utilisée pour la manipulation et l’animation des personnages virtuels. Le problème de la cinématique inverse peut être caractérisé par la détermination de la configuration de tout le corps du personnage virtuel ou de la configuration d’une partie de ce dernier à partir de la spécification de la trajectoire de l’extrémité.

2.5

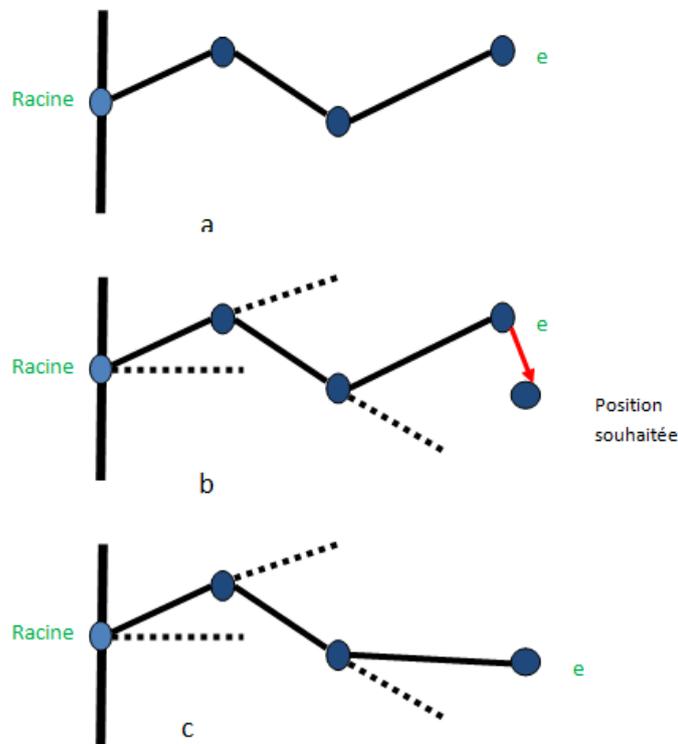


FIGURE 2.5 – Principe de la cinématique inverse : a-la premier figure tout en haut représente une chaîne cinématique d’un bras, b- Besoin de modification de la position de l’extrémité e afin d’atteindre une cible.c- En appliquant une méthode de la cinématique inverse nous avons obtenu notre but

D'après l'état de l'art donné dans le chapitre précédent, la cinématique inverse est l'une des méthodes la plus utilisée afin de modifier un mouvement en se spécifiant des contraintes. Chaque travail a utilisé une méthode différente de la cinématique inverse. Le choix de la méthode de la cinématique inverse est très important pour avoir des résultats naturels ou rapides selon la priorité de l'animateur.

2.5 Problèmes de la cinématique inverse

Dans cette section, nous allons nous concentrer sur le problème de la cinématique inverse, et les issues soulevées par sa résolution.

Essentiellement, la cinématique inverse est le problème de détermination de la configuration des jointures afin d'atteindre une tâche qui est exprimée dans l'espace cartésien. Par exemple, les configurations de l'épaule, coude et le poignet doivent être définies afin que la main atteigne une position dans l'espace. Les équations qui découlent de ce problème sont généralement non linéaires, et sont donc difficiles à résoudre. En plus, les techniques de résolution doivent traiter les difficultés décrites ci dessous.

2.5.1 Traitement de multiples contraintes et résolution de problème de conflit

La spécification d'une seule tâche, afin de contrôler des systèmes articulés très complexes n'est pas pratique. Alors qu'il n'est pas souhaitable qu'une technique gère de multiples tâches simultanément parce que cela peut provoquer des conflits où aucune tâche n'est satisfaite figure 2.6. Ce conflit doit être résolu par une stratégie appropriée.

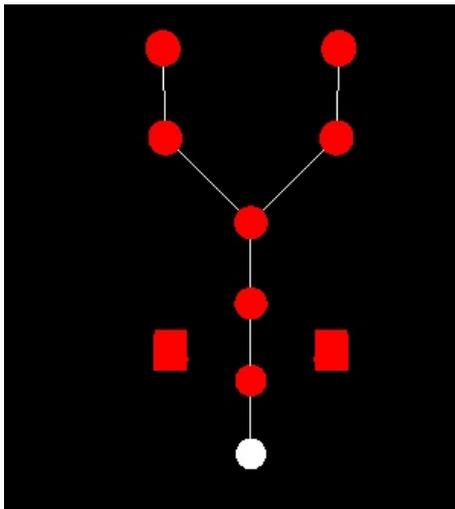
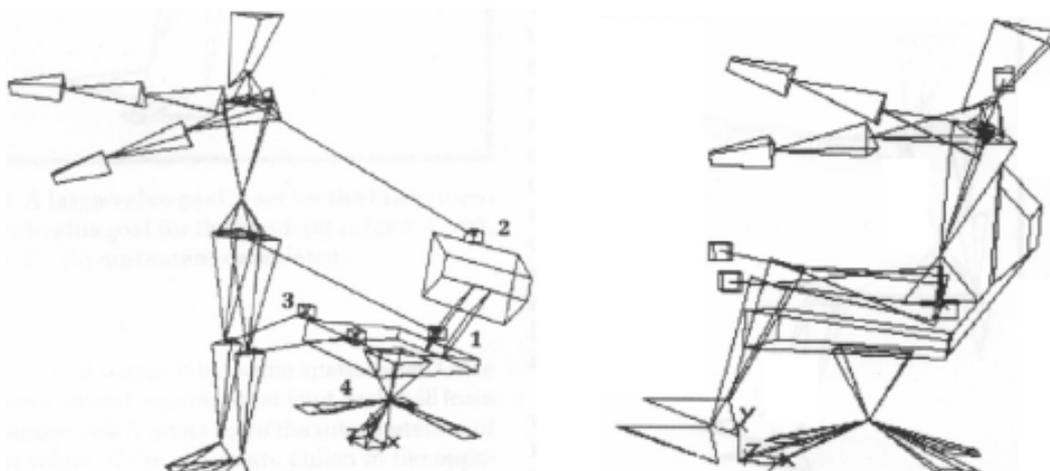


FIGURE 2.6 – Configuration simple : 8 articulations avec 2 cibles

Des solutions sont basées sur l'attribution des poids à chaque tâche [95, 116] et forment le problème comme une optimisation multi-objectif où aucune tâche n'est satisfaite mais la somme des erreurs est minimisée (figure 2.7). Alors que une stratégie est basée sur la priorité des tâches, ces dernières sont satisfaites selon l'ordre de priorité, la tâche la plus prioritaire est satisfaite en premier. Dans le travail [10], une méthode de la cinématique inverse numérique à base de priorité est proposée (PIK) elle se base sur un algorithme récursif, efficace et approprié aussi aux applications interactives. Raunhardt [85] a proposé une solution au problème dans l'espace latent en suivant le même principe de l'algorithme PIK, il a nommé son méthode LPIK.

FIGURE 2.7 – Positionnement d'un personnage dans une chaise en utilisant la stratégie de pondération. Différents poids sont attribués pour chaque tâche ($w_1=100, w_2=w_3=w_4=10$).

2.5.2 Traitement de Problème de sous contraintes et de sur contraintes

Nous discutons dans cette section le nombre de solutions du problème de la cinématique inverse. Trois situations peuvent se produire :

Le problème n'a pas de solution

Cette situation se produit lorsque la tâche ne peut pas être satisfaite, c-à-dire lorsque le but n'est pas atteignable, ou lorsque plusieurs tâches sont en conflit et ne peuvent pas être satisfaites simultanément, on appelle cela un problème sur-contraint figure 2.8.

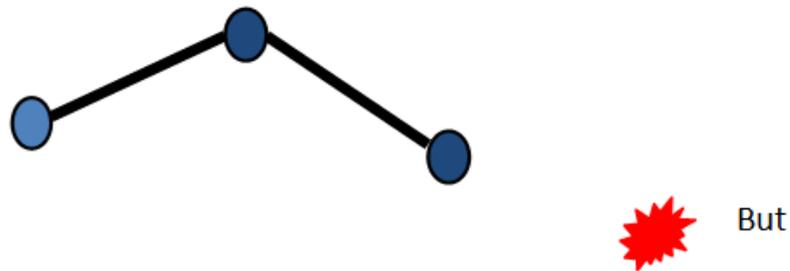


FIGURE 2.8 – Le problème n'a pas de solution (le but est loin)

Le problème a une seule solution

Ce cas ne soulève aucune problème, mais se produit rarement figure 2.9.



FIGURE 2.9 – Le problème a une seule solution

Le problème a plus de solutions

On dit qu'un système est redondant si le nombre de degrés de liberté est supérieur au nombre des contraintes à atteindre. le problème est appelé sous-contraint ou redondant. La structure est non redondante elle-même, mais seulement par rapport à une tâche qui nécessite moins de degrés de liberté que ceux disponibles dans la structure.2.10, 2.11

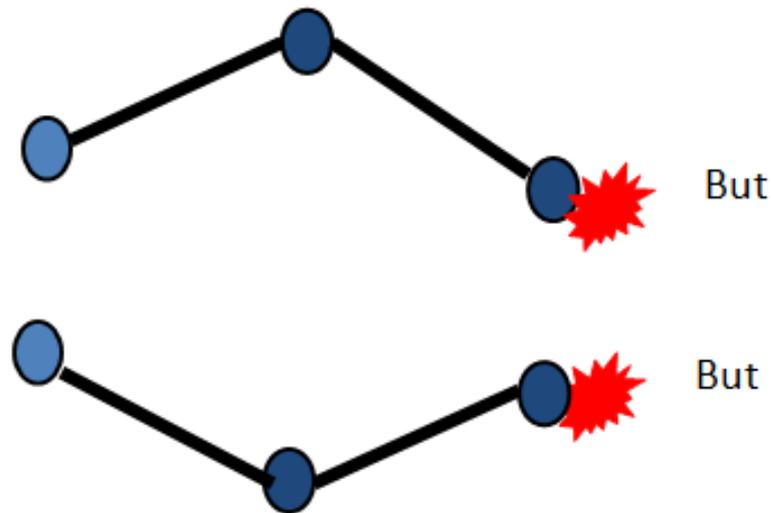


FIGURE 2.10 – Le problème a deux solutions : on peut atteindre l’objectif en appliquant les angles dans la figure en haut ou celle en bas



FIGURE 2.11 – le problème a une infinité de solutions : a chaque fois

une description mathématique de problème liée au nombre des résultats est donné dans la section 2.6.2.

2.5.3 Limites angulaires

Pour générer des mouvements similaires à ceux d’un être humain, un critère essentiel est le respect des limites angulaires, puisque leur violation peut affecter d’une manière significative la plausibilité de posture. C’est pourquoi n’importe quelle méthode de cinématique inverse pour les personnages virtuels doit avoir la possibilité de renforcer les angles de jointures selon les limites avant d’être considéré comme une solution viable. La méthode de résolution doit intégrer des mécanismes afin de renforcer précisément ces contraintes.

2.5.4 Obstacles

La majorité des environnements virtuels ne sont pas sans obstacles. Donc, la méthode de résolution de problème de la cinématique inverse doit gérer ce problème afin d'obtenir une posture sans que la jointure entre en collision avec les obstacles figure 2.12. Le problème d'évitement de collision a pour but d'assurer que les jointures ne sont pas en collision avec un objet de l'environnement lors du mouvement de l'extrémité du bras vers la cible (position finale).

Des solutions à base de priorité sont proposées, une priorité élevée est donnée afin de contraindre l'extrémité de suivre une trajectoire donnée, alors que la faible priorité est attribuée à la jointure qui est la plus proche de l'obstacle afin de l'éviter [72, 33]. Ces solutions sont basées sur la nature des chaînes redondantes, cette famille de travaux est efficace mais le temps de calcul de chaque itération est très élevé. Dans le domaine de l'animation, les méthodes à base de planification de chemin sont souvent utilisées, leur objectif consiste à atteindre un objet dans un environnement où il existe des obstacles [52, 50]. Dans cette classe, les résultats de la cinématique inverse sont renforcés par le fait qu'ils se trouvent dans le free-space (espace non obstacle), mais ces techniques sont très coûteuses en temps de calcul. Il sera aussi difficile à calculer le chemin dans les applications en temps réel. Pour les auto-collisions qui sont un cas spécial des collisions, Marcelo Kallmann [51] a présenté une méthode analytique « swing-and-twist parameterization » pour le mouvement des jambes et des bras en évitant les auto-collisions, cette méthode donne de bons résultats, néanmoins, elle ne fonctionne pas lorsqu'il existe des objets dans la scène.

Après cette description des problèmes liés à la résolution du problème de la cinématique inverse, nous allons présenter les différentes méthodes pour le résoudre, celles-ci prennent en charge les problèmes déjà cités.

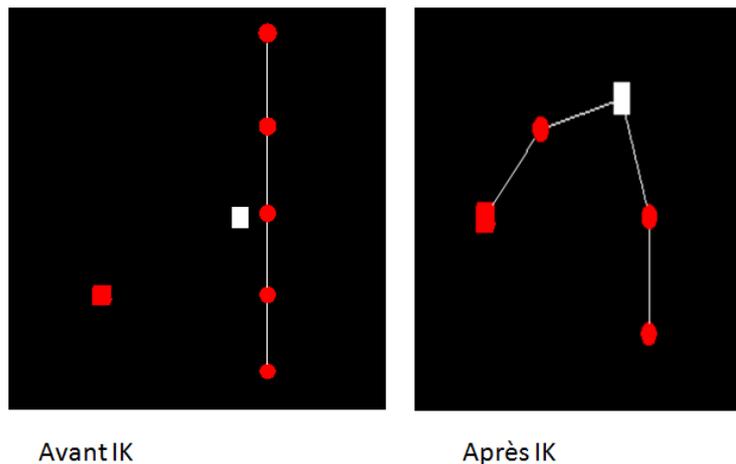


FIGURE 2.12 – L’extrémité atteindre l’objet mais sans prendre en considération l’obstacle.

Nous détaillons maintenant différentes méthodes utilisées pour résoudre le problème de cinématique inverse.

2.6 Techniques de la cinématique inverse

Dans cette section, nous allons donner un aperçu des méthodes les plus importantes à la résolution de problème de la cinématique inverse. Vu que le sujet est très vaste, la synthèse réalisée est loin d’être une étude exhaustive.

2.6.1 Méthode analytique

Le problème de la cinématique inverse peut être résolu de plusieurs manières. Les méthodes analytiques sont utilisées pour les chaînes cinématiques simples avec un nombre réduit de degré de liberté. Il consiste à modéliser le système articulé à l’aide d’un système d’équation. Le calcul de ce type de solution est très rapide et peut être utilisé dans les applications en temps réel. La solution peut être trouvée par la résolution directe d’un système non linéaire [84, 28].

Dans le domaine de l’animation, plusieurs chercheurs ont essayé de traiter le problème de manipulation des bras et des jambes. A titre d’exemple, Korein [59] présente une méthode analytique afin de résoudre le problème de la cinématique inverse pour les bras. La solution traite deux problèmes : le contrôle de la position en utilisant quatre degrés de liberté et le contrôle de la position/orientation utilisant sept degrés de liberté. Tolani et al. [99] combinent entre les solutions analytiques et numériques afin de résoudre le problème

de la cinématique inverse, la solution numérique est utilisée pour traiter le problème des limites angulaires. Menardais a utilisé une méthode analytique de la cinématique inverse afin de contraindre le personnage virtuel à toucher le sol [73].

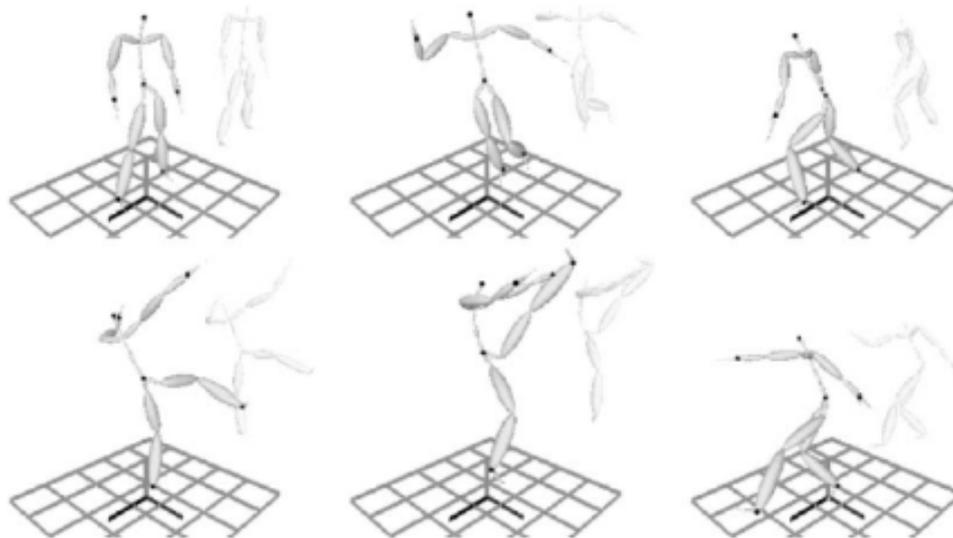


FIGURE 2.13 – Résultats de la méthode SIK, des échantillons d'animation : sur la gauche de chaque échantillon, la reconstruction de la posture utilisant SIK sur la droite c'est les postures originales [103]

Unzueta figure [103] 2.13 a proposé une méthode de la cinématique inverse séquentielle "Sequential Inverse Kinematics (SIK)", qui est une extension de la méthode présentée par Boulic [16]. C'est une méthode de cinématique inverse qui a comme but la reconstruction de la posture d'un personnage virtuel en utilisant un nombre réduit des jointures. La méthode reconstruit la posture en temps réel. Les entrées de cette méthode sont les positions des extrémités comme le poignet, pied, tête et le pelvis, le problème est ensuite résolu d'une manière séquentielle, en utilisant une méthode itérative de cinématique inverse analytique sur différentes parties du corps suivant un ordre définie. Selon Unzueta, la méthode SIK est plus performante que d'autres méthodes proposé par Kulpa [62] et Tolani [99]. Kallmann.[51] propose une méthode de la cinématique inverse analytique en intégrant une technique d'évitement d'obstacle. La méthode est personnalisée pour le contrôle souhaité afin d'atteindre des objets en temps réel.

Cependant, dans le cas d'un système plus complexe(plus d'articulations), ou dans le cas où l'on souhaite fournir une méthode fonctionnant pour tous les systèmes, la méthode analytique n'est pas adaptée. La méthode analytique est adaptée à un système en particulier, on doit recommencer si les articulations changent. Dans les faits, la plupart des

systèmes de cinématique inverse sont trop complexes pour être résolus au moyen d'une méthode analytique, les méthodes à base d'optimisation peuvent être utilisées dans ces cas afin de résoudre le problème de la cinématique inverse.

2.6.2 Méthodes à base d'optimisation

Cette classe de solveurs de problème de la cinématique inverse est formulée comme un problème d'optimisation à base de contraintes. La solution consiste à linéariser des équations non linéaires, la méthode la plus connue c'est "motion rate control" elle a été introduite par Whitney [110]. Les variations angulaires sont alors reliées à celle de l'extrémité par la relation :

$$\Delta X = J\Delta\theta \quad (2.3)$$

La solution a comme entrées la configuration initiale caractérisé par la matrice Jacobienne J de taille $M \times N$, cette matrice fait correspondre les variations des degrés de liberté μ des jointures en variation des paramètres de l'extrémité X . Elle est donc modélisé par une matrice de taille $M \times N$ qui rassemble les dérivées partielles des paramètres X de l'extrémité (au nombre de M) par rapport aux degrés de liberté μ (au nombre de N). Une fois la linéarisation résolue, la nouvelle configuration de l'extrémité qui est proche de la cible est calculée, en ajoutant la variation de vitesse de chaque jointure à la configuration initiale. Pour trouver Δ on peut exprimer l'équation de cette manière :

$$\Delta X = J^{-1}\Delta\theta \quad (2.4)$$

Dans le cas où $M = N$ et l'inverse du Jacobien existe, le système a un seul solution.

Dans le cas où $M = N$ et l'inverse de Jacobien n'existe pas, le système est singulier et il n'existe pas une solution.

Dans le cas où $M < N$ la matrice n'est pas carrée et l'inverse du Jacobien n'est donc pas défini dans ce cas le problème est dit "sous contraint". Il est possible de contourner la difficulté de l'inversion de la Jacobéenne en simplifiant le problème, en utilisant :

2.6.2.1 Pseudo-inverse

Introduite par Klein et Buss [56, 20] : on utilise une forme particulière de la matrice Jacobienne dite pseudo-inverse. Grâce à un algorithme de décomposition en valeurs singulières cette méthode présente l'avantage de fournir une solution de norme minimale

lorsqu'elle existe et sinon, une solution optimale au sens des moindres carrés. L'introduction de la pseudo inverse nous permet d'introduire une tâche secondaire, sans gêner la réalisation de la tâche principale, qu'est la cinématique inverse. Cette tâche secondaire correspond à une optimisation qui peut être de tout type. Cela donne l'équation suivante :

$$\Delta\theta = J^+\Delta X + \alpha(I + J^+J)\Delta Z \quad (2.5)$$

J^+ pseudo inverse de la matrice jacobienne J

I matrice identité

α constante de pénalité

ΔZ contrainte à minimiser qui correspond à la tâche secondaire.

2.6.2.2 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 [113] 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.

2.6.2.3 Tâches secondaire

Des solutions à base de Jacobinen sont utilisées aussi dans les cas de traitement de multiple tâches. Nakamura et al. [79] ont introduit l'idée d'exécution des tâches à base de priorité lorsqu'on traite les problèmes de la cinématique inverse. Similaires au travail de Nakamura, Maciejewski and Klein [72] ont proposé une formulation simple de problème de priorité des tâches. La tâche la plus prioritaire est utilisée afin de forcer l'extrémité de suivre un chemin prédéfinie. La tâche de basse priorité est crée pour contraindre le point qui est le plus proche de l'obstacle afin de gérer l'évitement d'obstacle. Siciliano and Slotine [95] généralisent le concept proposé par les deux auteurs, afin de traiter le problème de traitement du nombre arbitraire de tâches à base de priorité utilisant une formulation récursive. Baerlocher and Boulic [10] ont amélioré la généralisation du schéma de priorité en formulant une solution efficace et récursive qui accélère la convergence des performances. Afin de surmonter le problème de conflit lors de traitement de plusieurs tâches simultanément, des solutions sont basées sur l'attribution des poids à chaque tâche [8, 116] et formule le problème comme une optimisation multi- objectif où aucune tâche n'est satisfaite mais la somme des erreurs est minimisée.

Les méthodes à base d'optimisation donnent des résultats réalistes, mais avec un temps d'exécution très élevé. Récemment, une méthode basée sur la Jacobien destinée aux applications interactives a été introduite par Ken [54].

2.6.3 Méthodes à base d'heuristiques

Les méthodes à base d'optimisation donnent des résultats réalistes mais le temps de calcul est très élevé, de plus elles souffrent de problème de singularité due à l'utilisation des matrices. Afin de surmonter ces problèmes, des méthodes à base d'heuristiques sont proposées.

2.6.3.1 CCD (Cyclic Coordinate Descent)

La méthode CCD est une méthode itérative pour le contrôle des corps articulés [109, 105, 21], elle utilise moins de calcul numérique que la méthode du Jacobien inverse. Elle est implémentée dans la robotique, animation graphique et jeux vidéos. Son principe est de minimiser des erreurs de position et d'orientation en transformant une articulation chaque fois. Chaque itération parcourt la chaîne cinématique de l'extrémité à la racine et modifie chaque articulation séquentiellement. On effectue le processus tant que la distance entre l'articulation cible et sa position désirée n'est pas inférieure à un seuil.

Cette méthode est rapide mais elle n'est pas sans problème, l'inconvénient majeur de cette dernière c'est son non utilisation dans le cas de contraintes de plusieurs extrémités simultanément, Kulpa [62] a proposé une solution à ce problème en divisant le corps du personnage virtuel en groupes, chaque groupe est vu comme une chaîne cinématique individuelle. Ensuite, il a appliqué la méthode CCD sur chaque groupe séparément. Cette décomposition a l'avantage de faciliter le calcul de la solution. Néanmoins, par manque d'une relation entre les groupes, cela peut donner des résultats irréalistes.

Une extension de la méthode CCD est donné par Wang[105], c'est une méthode appelée *The Inductive Inverse Kinematics (IIK)*. Elle utilise la notion de *Uniform Posture Map (UPM)* afin de contrôler la posture du personnage virtuel. On peut aussi citer les travaux de Unzueta et al. [103] qui combinent la résolution analytique de colonne vertébrale et épaules avec une méthode CCD pour les membres qui restent (bras et jambes) pour reconstruire un mouvement à partir d'un nombre minimum de marqueurs sur le corps.

2.6.3.2 Triangulation

Une autre méthode heuristique qui s'appuie sur la triangulation à été présentée par Muler [76], cette méthode utilise la loi des-voisins *law of cosines*, afin de calculer l'angle de chaque jointure, elle démarre à partie de la racine de la chaine cinématique et se déplace à l'extérieur vers l'extrémité de la chaine. Cette méthode garantie l'existence trouver une solution lorsque les jointures sont sans contraintes et la cible est à portée.

La triangulation peut être appliquée uniquement pour les problèmes avec une seule extrémité [5], c'est pourquoi elle ne peut pas être utilisée pour le contrôle du personnage virtuel.

2.6.3.3 FABRIK : Forward And Backward Reaching Inverse Kinematics

D'autres méthodes proposent de résoudre le problème de cinématique inverse en se plaçant, non plus dans l'espace des orientations mais dans celui des positions (espace Euclidien). Brown et al [18] présentent une méthode non itératif en temps réel 'Follow-the-Leader' (FTL). Elle est proposée spécialement pour la simulation de la corde. Appuyant sur la même principe, Andreas à proposé sa méthode appelé FABRIK [5] 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 le suivis de la main. Andreas a comparé sa méthode avec d'autres méthodes de la cinématique inverse [5], 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 résoud 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. Le principe de fonctionnement de FABRIK est illustré dans la figure 2.14

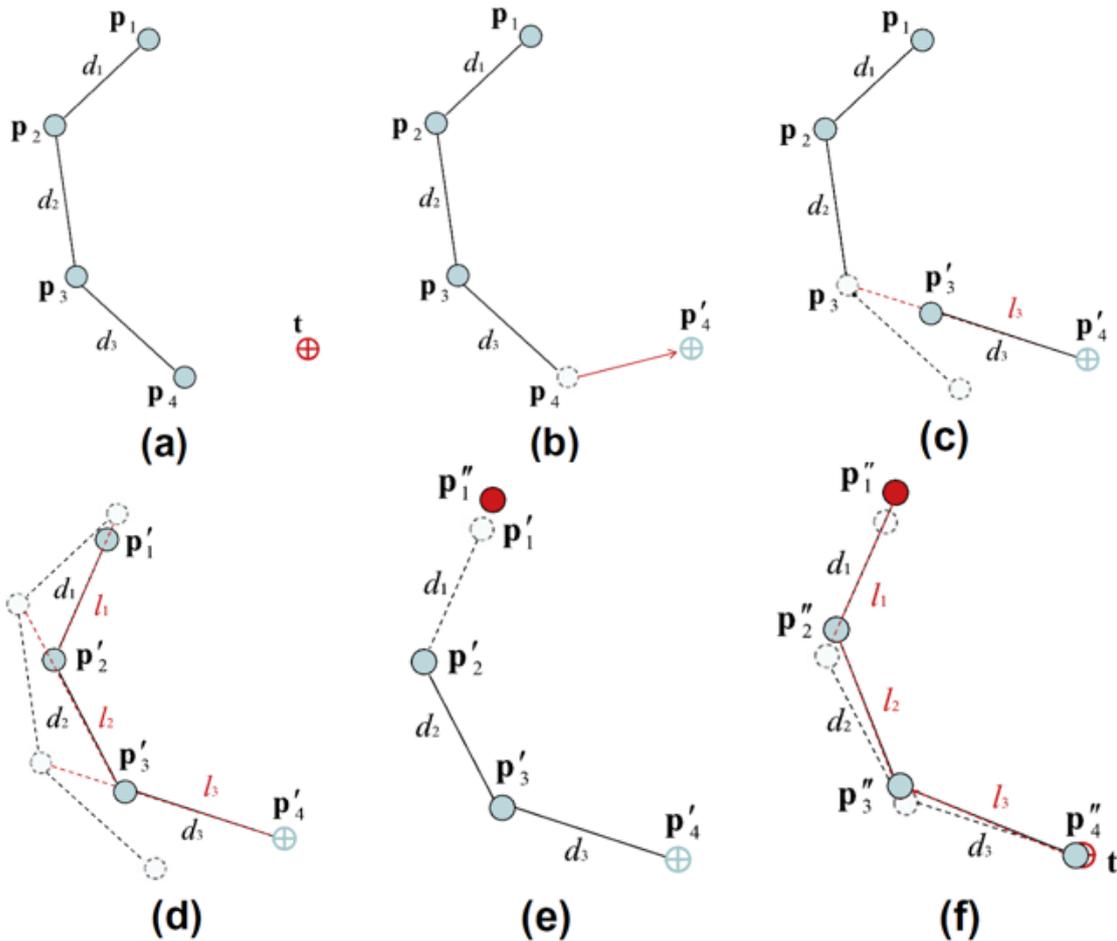


FIGURE 2.14 – Itération complète de FABRIK [5]

Huang et al [44], ont utilisé une variation de la cinématique inverse FABRIK afin de résoudre le problème de la cinématique inverse de point de vue du transfère d'énergie. Ils ont utilisés le modèle masse-ressort pour ajuster la position de jointure en minimisant l'énergie de la force qui est conservée dans la source.

Enfin, une méthode similaire à FABRIK, basée sur des contraintes de distance est proposée en robotique [40]. Cette méthode géométrique consiste à paramétrer le système par des distances inter-articulations, et à reformuler le problème à partir de la résolution d'un système d'inégalités linéaires.

Si la résolution utilisant FABRIK fournit une réponse au problème de cinématique inverse, elle présente certains inconvénients [12] comme :

- Le non respect de l'organisation segmentaire,
- L'existence des conflits lors de traitement de plusieurs tâches simultanément.
- L'absence des résultats satisfaisants en présence d'obstacles

2.6.3.4 Cinématique inverse à base d'exemple

L'idée de cette classe de méthodes est d'utiliser les mouvements déjà capturés afin d'aider la synthèse de la posture demandée, de telle sorte que des mouvements naturels peuvent être générés en utilisant un contrôl de signal de basses dimensions. Le comportement humain est directement intégré dans le processus de résolution du problème de la cinématique inverse où la posture finale préserve les caractéristiques (style, relation) de mouvement de la base.

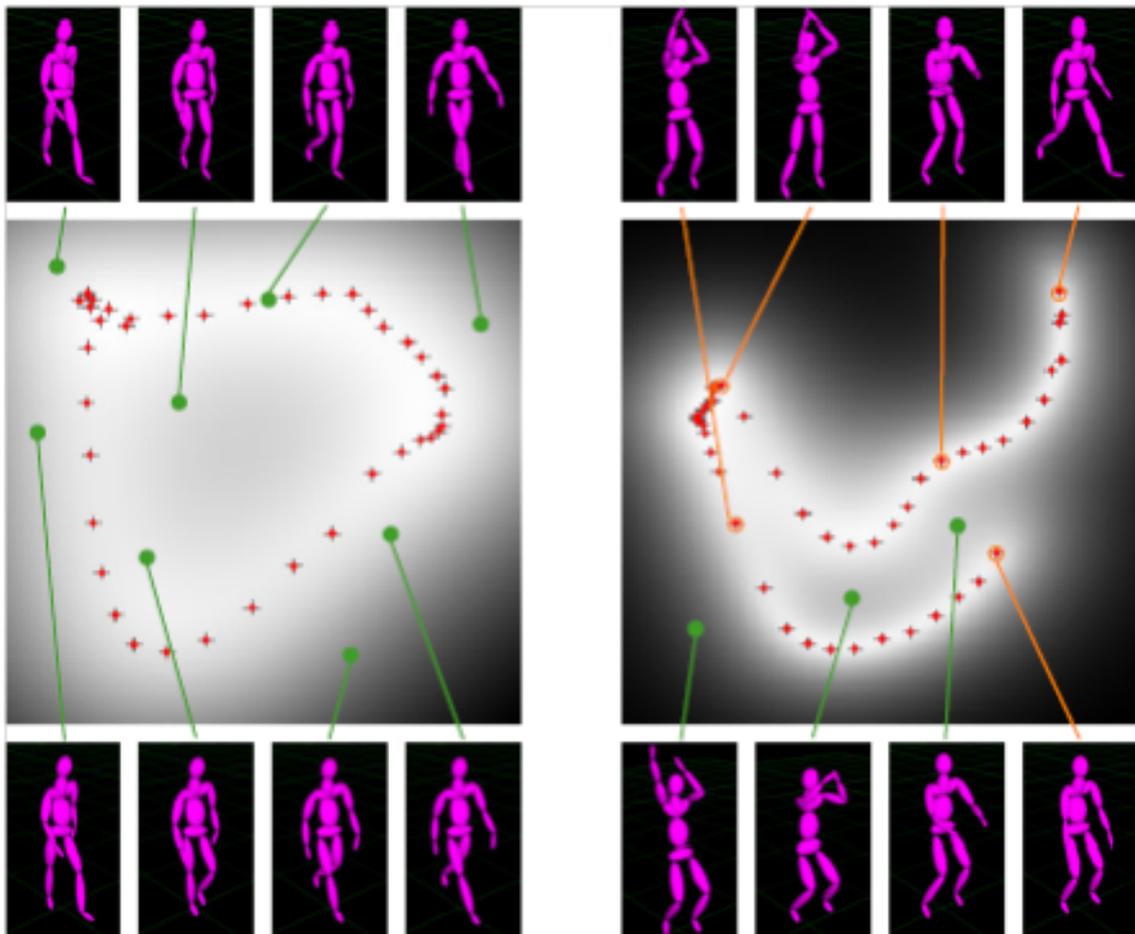


FIGURE 2.15 – Résultats de SGPLVM pour la marche [37].

Engin et Chen [30] ont proposé un modèle mathématique 3D de mouvement complexe d'un coude en se basant sur une base de mouvement. Ce modèle est capable de prédire la plage de mouvement de l'articulation du coude. Beck and Chaffin [11] ont proposé une approche de cinématique inverse comportementale développée en utilisant des équations de régression déjà existante. Verriest et al [104] ont utilisé un modèle statistique linéaire afin de prédire les postures de préhension pour un bras. Les deux approches déjà citées utilisent

des équations de régression statistiques développées à partir d'une base de données qui contient des mesures sur le comportement de l'humain. Dans le but de positionner le bras du personnage virtuel en précisant la position souhaitée de la main, Wang and Verriest [107] ont proposé un algorithme de cinématique inverse géométrique spécifique pour les bras du personnage virtuel qui intègre les limites angulaires de l'épaule.

Grochow et al. [37] ont proposé un solveur de cinématique inverse à base de style *style-based IK (SIK)* figure 2.15, afin de modéliser les mouvements humains, ils ont appliqué une technique de réduction de dimensions non linéaires appelée *Scaled Gaussian Process Latent Variable Model (SGPLVM)*. L'utilisateur utilise la souris afin de contrôler quelques articulations, ensuite le système cherche les postures appropriées qui satisfont les contraintes dans l'espace latent. A part SGPLVM, la modélisation de mouvement qui se base sur la réduction de dimensions est très populaire. Par exemple [17, 68] se sont basés sur le ACP, ils ont combiné un modèle d'apprentissage de probabilité avec des contraintes cinématiques afin de contrôler interactivement le personnage virtuel avec la souris.

Rose et al [87] Ont proposé un solveur de cinématique inverse à base d'exemples, qui peut résoudre le problème de la cinématique inverse efficacement en cherchant et interpolant les mouvements appropriés. Pour traiter le problème les bases de mouvement larges, Wu et al [114] ont proposé en plus le concept des postures représentatives *representative postures*, qui est une partie des postures spéciales dans la base de mouvement. Cependant, le coût de l'apprentissage et la complexité de système augmente avec la quantité de la base de mouvement, et l'efficacité de réduction de dimension est réduite avec l'augmentation de la diversité de la base de mouvement

Carvalho et al [22] Ont combiné un modèle de mouvement ACP avec un framework PIK, afin de résoudre des contraintes externes dans l'espace de bas niveau. Le solveur PIK traite les problèmes de déformations de quelques caractéristiques dans l'espace de mouvement ACP. L'utilisation de l'espace de basse dimension augmente la possibilité d'exécuter la méthode IK dans l'espace des mouvements possibles. Etant donnée que la dimension de l'espace du mouvement est petit en comparaison à celui de l'espace d'articulation d'un mouvement, LPIK a l'avantage de réduire la taille de la matrice jacobienne. La méthode offre l'avantage d'être bien adaptée à la gestion de personnage avec un nombre important de degrés de liberté.

Contrairement aux approches à base d'apprentissage hors ligne, la modélisation temps réel a montré qu'elle est efficace pour les applications en temps réel avec une large base de mouvements. L'idée est de sélectionner un nombre réduit de postures basant sur les

informations au cours d'exécution afin de synthétiser la posture demandée. Une méthode naïve consiste à chercher dans la base la meilleure posture pendant l'exécution en se basant sur les entrées de l'utilisateur afin de synthétiser le mouvement du personnage [94]. Une autre approche applique une Analyse en Composantes Principales locale est proposée par Chai et Hodgins [23] elle construit l'espace latent au cours de l'exécution [23]. Des postures qui sont similaires à la posture actuelle sont utilisées afin de construire l'espace latent de telle sorte qu'un mouvement peut être synthétisé en utilisant un nombre réduit de contraintes sur les positions. Liu et al [70] ont étendu l'idée en utilisant « maximum a posteriori framework » afin de reconstruire le mouvement, ce qui renforce la cohérence temporelle du mouvement.

Country [27] et Hecker [41] ont proposé une méthode séquentielle à base de monte-Carlo SMCM et un filtrage à base de particule respectivement. Les deux méthodes donnent des bons résultats avec l'élimination de problème de la singularité, mais le problème de temps de calcul élevé est toujours présent.

Le problème général de ces méthodes est la difficulté d'assurer que les postures extraites sont similaires à celle des métriques cinématiques utilisés. Les postures des différentes topologies peuvent donner des résultats irréalistes avec des auto-collisions.

Les méthodes à base d'exemples peuvent générer n'importe quelle posture, mais les postures sont très liées, à celles qui sont similaires à l'espace des postures dans la base de l'apprentissage.

2.7 Bilan

La cinématique inverse est utilisée afin de contrôler les postures des figures articulés. 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 2.7 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 Ik	Simplicité	rapidité	continuité	Autres caractéristiques
Jacobien [110]	- -	- - -	+	Problème de singularité
CCD [109]	+	+	-	Manque de réalisme
SIK [103]	+	-	-	Problème de singularité
Triangulation [18]	-	+ +	-	Pour une seule extrémité
FABRIK [5]	+ +	+ +	+ +	Nécessite des adaptations au cas des personnages

TABLE 2.1 – Comparaisons des principales méthodes de cinématique inverse

Les approches utilisant le Jacobien, le temps de calcul nécessaire est beaucoup plus important car il faut calculer la jacobienne puis l'inverser soit avec la pseudo-inverse soit avec la transpose. Les méthodes à base de CCD nécessitent un temps de calcul faible. Cependant, le nombre d'itération pour parvenir à la solution est généralement très élevé qui peut alors affecter la qualité de résultat [109]. Le système de contrôle analytique proposé par de Unzueta [103] est simple, de nouveau, le système est limité à contrôler quelques membres (la colonne vertébrale et les clavicules sont traitées separement).

Dans un autre coter, l'évitement d'obstacle est l'un des problèmes que les méthodes de cinématique inverse rencontré lors de leur utilisation dans le domaine de modification de mouvement. Le tableau2.7 présente un bilan sur les principaux travaux qui servent à traiter ce problème. On peut conclure que les méthodes qui se basent sur la nature de la cinématique inverse sont efficaces et donnent des résultats dans des temps raisonnable par rapport à celle qui se base se la planification des chemins.

Famille	Méthode	Caractéristiques
À base de la cinématique inverse	Maciejewski and Klein [72], Kallmann [51]	Efficace mais le temps de calcul est très élevé
Planification des chemins	Marcelo Kallmann [52, 50]	<ul style="list-style-type: none"> – Renforcer les résultats de la cinématique inverse d’être dans le free space en utilisant les techniques de planification de chemin – Très utile dans le domaine d’animation – Très couteux en temps de calcule – Difficile à calculer le chemin dans les applications interactive

TABLE 2.2 – Caractéristiques des méthodes principales d’évitement d’obstacle

2.8 Conclusion

Dans les sections précédentes, 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 la 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 interactive, il est souvent conseillé de choisir une méthode rapide.

Une méthode basée sur l’algèbre géométrique est récemment proposé par Andrias [5], elle utilise des lignes et des sphères afin de résoudre le problème de la cinématique inverse. Néanmoins elle présente aussi des problèmes lors de l’utilisation pour les personnages virtuels.

Dans le chapitre suivant, nous présenterons notre méthode de la cinématique inverse utilisée pour la modification de mouvement et qui répond à nos exigences. Dans le chapitre suivant, nous présenterons notre proposition de la cinématique inverse, qui a pour but principal la modification des mouvements en répondant à nos exigences.

PIKOV : Cinématique inverse avec évitement d'obstacles et multiples contraintes avec priorité

3.1 Introduction

Le contrôle des systèmes articulés est très demandé dans plusieurs domaines tels que la robotique et l'animation. Dans l'animation, le contrôle sert à positionner le corps du personnage virtuel afin de respecter des contraintes cinématiques ou dynamiques. L'objectif de cette contribution est de proposer une nouvelle méthode de contrôle cinématique des systèmes articulés "personnage virtuel" dont le but est de modifier un mouvement pour respecter de nouvelles contraintes.

3.2 Problématique

Au niveau de notre travail, nous nous sommes intéressés à la manipulation et l'ajustement d'une posture existante d'un personnage virtuel, en utilisant la cinématique inverse.

La méthode de la cinématique inverse utilisée pour atteindre notre objectif doit répondre aux exigences ci-dessous :

- Rapidité : la méthode de modification de mouvement proposée doit supporter l'aspect d'interactivité, où l'utilisateur a la possibilité de contrôler les jointures du personnage virtuel en utilisant des dispositifs externes tel que la souris. Ce qui nécessite la rapidité de la méthode.

- Simplicité : l'utilisation d'une méthode simple qui n'utilise pas par exemple les matrices augmente la rapidité de la méthode. En plus, la simplicité de la méthode offre l'avantage de son utilisation dans différents systèmes. Un autre avantage c'est l'ajout des nouvelles contraintes s'effectue d'une façon facile.
- Réalisme : la posture finale de la cinématique inverse doit être naturelle parce que l'objectif est d'avoir des postures similaires à un mouvement d'une vraie personne.
- Contraintes multiples : la méthode de la cinématique inverse proposée doit supporter de contraindre plusieurs jointures en même temps et sans conflit.
- Évitement d'obstacles intégrés dans la méthode IK : dans ce point, il y a deux aspects, le premier, l'environnement peut contenir des objets, c'est pourquoi la méthode de la cinématique inverse doit avoir une stratégie d'évitement d'obstacles. Le deuxième aspect, consiste d'intégrer la stratégie d'évitement d'obstacles dans l'algorithme de la cinématique inverse. Pour expliquer l'avantage de l'intégration de la stratégie dans l'algorithme de la cinématique inverse, supposons que nous avons appliqué la cinématique inverse, nous obtenons comme résultat l'extrémité dans la position souhaitée mais avec une collision entre une jointure et un objet. Ensuite, en appliquant l'algorithme d'évitement d'obstacles, cela peut nuire aux contraintes vérifiées par la cinématique inverse. Pour tel raison, la stratégie d'évitement d'obstacles doit être intégrée dans l'algorithme de la cinématique inverse.

Dans ce chapitre, nous décrivons la méthode de la cinématique inverse proposée qui essaye de répondre à ces exigences, ensuite nous validons nos résultats par une comparaison entre la méthode proposée et d'autres méthodes avec le même principe de base.

3.3 PIKOV : *Priority Inverse Kinematic with Obstacle Avoidance*

Notre méthode de la cinématique inverse a le même principe de base que FABRIK [5], elle est donc basée sur le principe d'algèbre géométrique et elle s'effectue en deux étapes forward et backward. La nouvelle position de jointure est calculée en gardant la distance entre $joint_i$ et $joint_{i-1}$.

Cependant, nous intégrons un algorithme afin de traiter le problème des obstacles ainsi que les conflits entre les tâches. Dans les sous sections suivantes nous présentons notre idée afin de résoudre le problème de cinématique inverse tout en prenant en compte les

obstacles et les conflits entre les tâches simultanées.

3.3.1 Évitement d'obstacles

La cinématique inverse est largement utilisée pour l'édition des mouvements des personnages virtuels. Par contre, la plupart des méthodes existantes éditent directement les rotations ou les positions des jointures, sans prendre en considération l'environnement du mouvement [62, 22].

Dans cette étape, l'objectif est d'assurer que les jointures du personnage n'entrent pas en collision avec les objets statiques et dynamiques dans l'environnement (obstacles). Les obstacles statiques représentent les objets qui ne bougent pas, alors que les obstacles dynamiques représentent les objets qui changent de position au cours du temps. La stratégie d'évitement de collision doit être intégrée dans l'algorithme de la cinématique inverse.

Notre proposition est basée sur l'algèbre géométrique « GA » afin de calculer les nouvelles positions des jointures sans collisions figure 3.1. L'avantage de l'intégration de l'algorithme d'évitement d'obstacles dans la méthode IK est que l'application de celui-ci après la méthode de la cinématique inverse peut nuire aux contraintes vérifiées par cette dernière. Tandis que l'on se base sur l'algèbre géométrique pour proposer un algorithme d'évitement d'obstacle engendre l'avantage de garder le même principe de base de la méthode de la cinématique inverse (algèbre géométrique) et par conséquent la cohérence de la méthode de la cinématique inverse est garantie. Au plus, garder la rapidité de l'algorithme de la cinématique inverse, donc la possibilité de son utilisation dans les applications interactives.

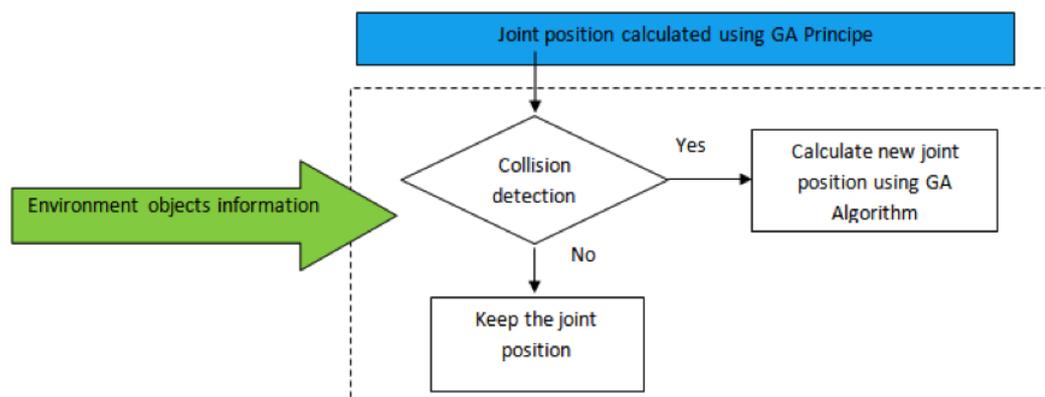


FIGURE 3.1 – Stratégie d'évitement d'obstacles

Deux étapes principales sont proposées au sein de notre algorithme : la détection et

l'évitement de collisions figure 3.2. Nous appliquons cet algorithme à chaque frame ce qui donne aussi de bons résultats pour les objets dynamiques .

3.3.1.1 Détection des collisions

Dans cette étape, nous vérifions l'existence des collisions entre les jointures et les objets de l'environnement. Pour réaliser cet objectif, nous utiliserons les boîtes englobants *axis aligned bounding boxes (AABB)* [13], qui permet d'englober les objets de l'environnement avec des formes géométriques simples (généralement des cubes ou des sphères), ensuite, nous testons si une jointure est en intersection avec une AABB ou non. L'utilisation des AABB simplifie le problème d'intersection des formes complexes. Un autre avantage d'utilisation des AABB c'est que l'algorithme est indépendant de la forme des objets.

3.3.1.2 Évitement des collisions

Nous nous sommes basés dans notre travail sur une méthode locale de la cinématique inverse. Dans cette méthode, la nouvelle position de jointure dépendant de la position précédente de la même jointure. Dans notre proposition, l'algorithme agit différemment afin d'éviter la collision : la nouvelle position d'une jointure ne dépend pas seulement de la position déjà calculée de la jointure mais dépend aussi des positions des obstacles.

Dans l'algorithme de la cinématique inverse avec évitement d'obstacles PIKOV, nous calculons la nouvelle position $joint_i$ pendant que nous gardons la distance D entre $joint_i$ et $joint_{i-1}$. Nous définissons pos_i comme étant la nouvelle position. Si $joint_i$ dans la position $pos_i(x_0, y_0, z_0)$ entre en collision avec un objet, nous calculons une nouvelle position $pos'_i(x, y, z)$ pour cette jointure, sinon, nous gardons cette position et nous passons pour le calcul de la nouvelle position de la jointure suivante. pos'_i doit respecter deux distances : la distance D et $(dis_{max} + \Delta d)$. dis_{max} est la distance entre $joint_i$ et les bordures du volume englobant et Δd est une valeur de sécurité afin d'assurer que la jointure dans cette nouvelle position n'entre pas en intersection avec les objets de l'environnement (Δd choisi 0.01). Afin de respecter les deux distances, nous utilisons les notions d'algèbres géométrique (GA). Pour calculer la nouvelle position d'une jointure dont on connaît les positions $joint_i$ et $joint_{i-1}$, en utilisant l'algèbre géométrique, le problème se reformule aux intersections de deux sphères. La première sphère S est caractérisée par son centre $pos_i(x_0, y_0, z_0)$ et son rayon dis ($dis = dis_{max} + \Delta d$). La deuxième sphère $S2$ est caractérisée par un centre $joint_{i-1} (x_1, y_1, z_1)$ et un rayon D . L'intersection des deux sphères

implique la résolution d'un système d'équations (3.1).

$$\begin{aligned}(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 &= dis^2 \\ (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= D^2\end{aligned}\tag{3.1}$$

La solution de ce système d'équations (3.1) est un cercle. Comme on a besoin de trouver un seul point qui sera la nouvelle position de jointure $joint_i$ sans collision, On propose de fixer la coordonnée Z. Afin d'accomplir cette étape, et vue que nous avons une somme égale à D^2 où dis^2 , $(z - z_0)$ ou $(z - z_1)$ ne doivent pas dépasser le rayon . On utilise l'équation 2 afin de fixer z

$$|z - z_1| < Det |z - z_0| < dis\tag{3.2}$$

Pour trouver la nouvelle position $pos'_i(x, y, z)$ de $joint_i$ ayant comme position actuelle $pos_i(x_0, y_0, z_0)$ on doit résoudre le système d'équations suivant :

$$\begin{aligned}(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 &= dis^2 \\ (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= D^2 \\ |z - z_1| < D \text{ et } |Z - z_0| &< dis\end{aligned}\tag{3.3}$$

L'algorithme proposé pour trouver la nouvelle position sans collision avec un objet de l'environnement est résumé dans ce pseudo :

Algorithme 1 Évitement d'obstacles

- 1: Entré : position de $joint_i$: $Pos_0(x_0, y_0, z_0)$, liste des objets, $pos_{i-1}(x_1, y_1, z_1)$
 - 2: Sortie : nouvelle position de P_1 : $pos'_i(x, y, z)$
 - 3: Vérifie l'intersection entre $joint_i$ et les objets de l'environnement
 - 4: S'il n'y a pas une intersection
Ne fait rien et garde $pos'_i(x, y, z)$ comme $joint_i$ position: $Pos_0(x_0, y_0, z_0)$
 - 5: Sinon
 - 6: Calculer les distances entre $pos'_i(x, y, z)$ et les coins de la boîte englobant ($d1....dn$)
 - 7: Calculer la distance $dist_{max} = max(d1....dn)$
 - 8: Calculer z en utilisant l'équation 3.3
 - 9: Calculer x, y en résolvant le système d'équations 3.1 et 3.2;
 - 10: Fin
-

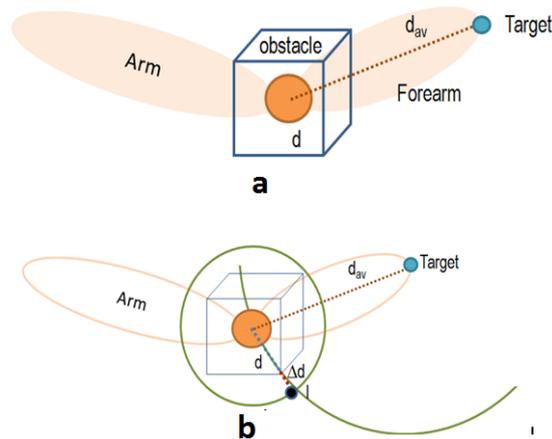


FIGURE 3.2 – Évitement d’obstacles en utilisant l’algèbre géométrique (GA) : le coude entre en collision avec l’objet (a), après l’application de l’algorithme d’évitement d’obstacle le coude n’est pas en collision avec l’objet (b)

3.3.2 Multiples tâches

Il est essentiel pour un solveur de problème de la cinématique inverse de résoudre des problèmes avec de multiples tâches. Cependant, sans une bonne stratégie, cela peut conduire à des conflits entre les tâches. Dans notre proposition et afin de résoudre ce problème en se basant sur le travail de Boulic [10], nous avons proposé de donner une priorité à chaque tâche.

Dans le cas où nous souhaitons contraindre plusieurs jointures, il est important de satisfaire premièrement les contraintes prioritaires avant les autres. En effet, si l’utilisateur veut attraper un objet avec sa main sans lâcher un autre objet avec l’autre main, il faut donner plus d’importance à la première contrainte qu’à la deuxième. Cette importance ou priorité est spécifiée par un paramètre $priority_i$ pour une $tche_i$ qui est une valeur positive, d’ailleurs, il existe deux grands modes de contrôle des contraintes en fonction de ces priorités : contrôle par priorité et par mélange. Afin d’offrir plus de fonctionnalités aux utilisateurs, notre travail est compatible avec ces deux modes de contrôle. Avec les mêmes contraintes, l’utilisateur peut donc décider de choisir le mode de contrôle souhaité au moment de l’exécution.

3.3.2.1 Contrôle par priorité des contraintes

Dans le travail [9], les contraintes sont respectées selon l’ordre de leur priorité. Ainsi la contrainte de la plus forte priorité est respectée la première, ensuite la contrainte suivante

est tentée d'être vérifiée sans mettre en cause la précédente, cette « adaptation » est répétée pour toutes les contraintes. Kulpa et al [62] aussi dans leur travail, essayent de respecter les contraintes de plus fortes priorités et s'approchent au mieux des suivantes, les contraintes sont triées dans l'ordre décroissant de leur priorité.

Dans notre travail et afin de satisfaire la tâche la plus prioritaire, nous effectuons un tri dans l'ordre croissant des tâches selon leurs priorités. Nous obtenons plusieurs chaînes simples, ensuite nous appliquons la méthode de la cinématique inverse pour chaque chaîne simple, en démarrant par la tâche la moins prioritaire vers la plus forte. Le démarrage par la faible priorité nous assure que la forte priorité sera atteinte.

3.3.2.2 Contrôle par mélange des contraintes

Lorsqu'une somme pondérée de toutes les contraintes est effectuée, le résultat est une somme pondérée des adaptations de chacune des contraintes prises séparément.

Dans le travail de Kulpa [62], la somme pondérée est celle des rotations, alors que dans notre travail et vu que notre algorithme de la cinématique inverse calcule des positions et que notre objectif est d'intégrer le module de traitement des conflits entre les tâches dans le même algorithme, la somme pondérée concerne des positions et non pas des rotations.

Pour calculer la nouvelle position de jointure qui relie les chaînes simples, l'algorithme calcule en premier les positions pos_i correspondant aux positions satisfaisant de chacune des contraintes de façon isolé. Ensuite, la somme pondérée de ces positions est calculée afin d'obtenir la position finale $newpos$ à appliquer au jointure de relation.

$$newpos = \frac{\sum_{i=1}^{i=n} pos_i \times priority_i}{\sum_{i=1}^{i=n} priority_i} \quad (3.4)$$

Pos_i est la position de l'extrémité i , $Priority_i$ est la priorité de la tâche à atteindre par l'extrémité i .

Afin de mieux expliquer l'algorithme, supposons que l'*extrmit* _{i} doit atteindre la cible ayant une priorité $priority_i$, lorsqu'on fait appel à l'étape forward pour toute la chaîne de i , on obtiendra une nouvelle position de ces jointures de relation pos_i . A la fin de cette étape, plusieurs positions de sub-base sont obtenues, la nouvelle position de sub-base représente la moyenne de toutes les positions pondérées par les priorités des tâches correspondantes, ce qui permet de tirer la nouvelle position vers l'extrémité la plus prioritaire et satisfaire sa tâche. La nouvelle position est donnée par la formule 3.4.

La figure 3.3 illustre l'utilisation de ces deux modes de contrôle. Soit une chaîne com-

3.3. PIKOV : *PRIORITY INVERSE KINEMATIC WITH OBSTACLE AVOIDANCE*

posée de 9 maillons et de 10 jointures, supposée représenter le rachis et les membres supérieurs. Nous avons placé deux contraintes afin d'atteindre deux objets par les membres supérieurs. Les deux sphères en vert représentent les cibles à atteindre.

La partie (a) représente l'application de la cinématique inverse sans l'utilisation de la notion de priorité. La partie b représente l'utilisation de mode par priorité des contraintes, la cible de la partie droite est plus prioritaire et par conséquent complètement vérifiée, tandis que l'autre cible est respectée au mieux. La partie c représente le contrôle par mélange des contraintes.

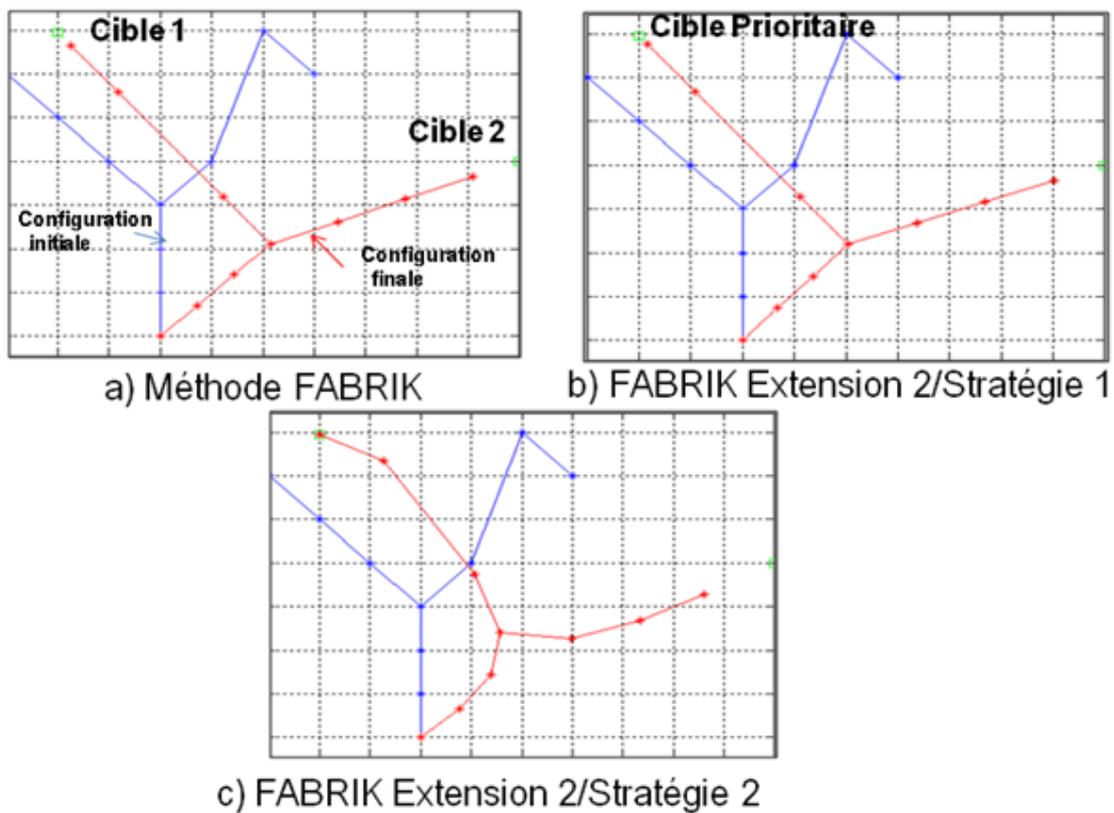


FIGURE 3.3 – Atteint de deux contraintes simultanément : Deux contraintes C1 et C2 sont appliquées sur les deux mains. Les sphères représentent les cibles de ces contraintes. La priorité des contraintes respecte toujours la contrainte de plus forte priorité

Comme illustrée dans la partie (a). La partie (b) montre le mélange des contraintes avec un poids de valeur 4 pour la main droite et 1 pour la gauche. La partie (c) utilise cette fois des poids de même valeur pour les deux contraintes sur les poignets.

3.3.3 Limites angulaires

Pour générer des mouvements similaires à ceux d'une personne réelle, un critère essentiel est le respect des butées articulaires- limites angulaires-. En effet, chaque mouvement permet un débattement articulaire limité que nous prendrons en compte dans notre méthode de la cinématique inverse.

Il convient de noter que les amplitudes articulaires du buste sont variables suivant l'âge, elles diminuent en vieillissant, et suivant l'individu, selon sa souplesse. Il est possible d'imposer d'autres valeurs de butées articulaires à notre algorithme pour simuler également des personnes à mobilité réduite ayant des débattements articulaires réduits.

3.4 Résultats et comparaison

A cause de l'absence d'une méthode qui se base sur l'algèbre géométrique et qui a le principe de fonctionnement de notre méthode de cinématique inverse et qui gère le problème d'évitement d'obstacle et de traitement de conflits lors de l'exécution de plusieurs tâches simultanément, nous allons comparer notre méthode avec une méthode qui a le même principe de fonctionnement. Une autre comparaison est faite avec la méthode connue "PIK" qui gère le problème de conflit entre les tâches.

3.4.1 Comparaison avec FABRIK

La méthode FABRIK a le même principe de fonctionnement que notre méthode qui est l'algèbre géométrique. Dans les sections suivantes nous comparons nos résultats avec ceux de la méthode FABRIK.

3.4.1.1 Multiple tâches

Nous avons implémenté la méthode FABRIK avec multiples contraintes et notre méthode de cinématique inverse PIKOV.

La figure 3.4 compare l'utilisation des deux stratégies proposées (par priorité et par mélange) sur une chaîne articulée avec deux extrémités pour l'utilisation de deux contraintes ? une pour chaque extrémités, dans le cas des personnages virtuels ces chaînes représentent le tronc et les bras. . Dans l'intérêt de cet exemple, les positions des cibles ont été choisies de manière à ce quelles soient accessibles par les extrémités une par une, mais pas les deux en même temps. Dans le premier cas (a) un algorithme de cinématique inverse FABRIK

3.4. RÉSULTATS ET COMPARAISON

est appliqué. Nous remarquons que les extrémités sont placées à distances en direction des cibles sans pour autant les toucher. Dans le second cas (b), nous pondérons la priorité des tâches. Ainsi, la cible de gauche ayant une plus forte priorité, la distance entre cette cible et l'extrémité va être inférieure à la distance entre l'autre cible et l'extrémité. Dans le dernier cas (c), la cible de gauche possède une priorité plus élevée que la cible de droite. Ainsi une extrémité est bien amenée sur la cible gauche, tout en essayant de toucher la cible droite? sans pour autant y arriver.

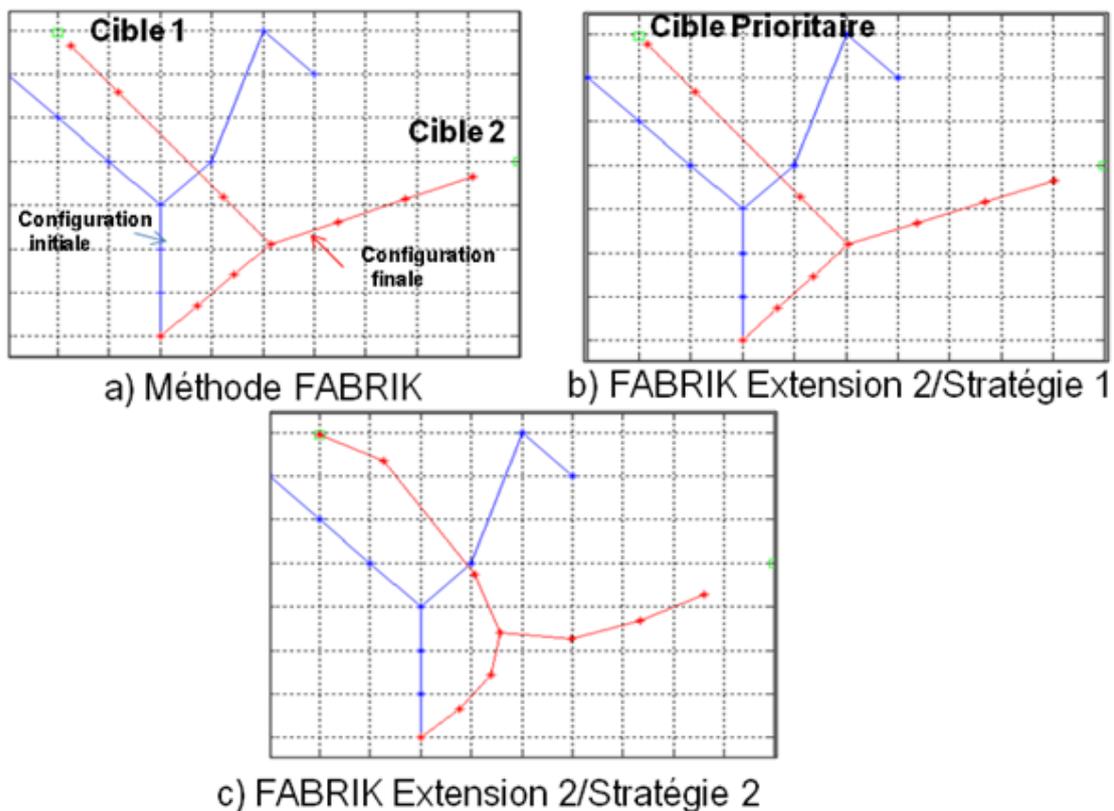


FIGURE 3.4 – Cas de multiple tâches

Les résultats du contrôle par mélange de priorités dépendent des valeurs de la priorité de chaque tâche, le choix de ces priorités est très important, un mauvais choix peut atteindre la tâche la plus prioritaire mais en temps de calcul plus élevé. Alors que les résultats de la deuxième stratégie dépendent uniquement de la tâche la plus prioritaire.

Comme conclusion, la deuxième stratégie est utilisée lorsque la tâche doit être atteinte impérativement. Il est aussi possible de contraindre n'importe quel point du squelette, pas seulement deux points.

Nous avons testé l'algorithme sur trois tâches.

3.4. RÉSULTATS ET COMPARAISON

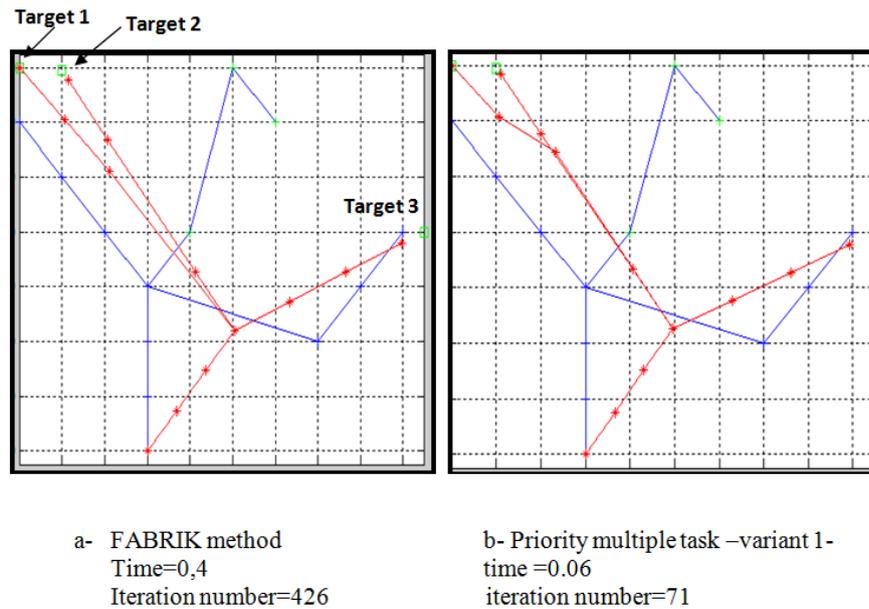


FIGURE 3.5 – Atteint de trois contraintes simultanément : utilisation IK avec priorité comparer à FABRIK, nous avons besoins de moins temps de calcul et peu d’itération afin d’atteindre les tâches.

Dans la figure4.6, nous avons testé et comparé la méthode FABRIK lors du traitement de trois tâches avec notre méthode. Nous avons obtenu des résultats améliorés, en effet Lors de l’application de la méthode FABRIK avec trois extrémités figure4.6.a, le nombre d’itérations afin d’atteindre les cibles est très élevé. Dans la figure 4.6.b, après l’application de notre méthode à base de priorité, le temps de calcul et le nombre d’itérations diminuent. FABRIK a besoin de 426 itérations et de 0.4s pour atteindre les 3 cibles, alors que il y a deux tâches ne sont pas atteinte et un seul qui l’est, alors que lors de l’utilisation de notre méthode on n’a besoin que de 71 itérations et 0.06 s pour atteindre les 2 cibles les plus prioritaires.

3.4.1.2 Évitement d’obstacles

Dans cette section et pour prouver l’efficacité de notre méthode d’évitement d’obstacle La figure 3.6-a représente une chaîne cinématique dans un état initial dans un environnement où il y a des obstacles et un objet à atteindre. En appliquant la méthode FABRIK figure 3.6-b l’extrémité atteint la cible mais des articulations de la chaîne sont en collision avec d’autres objets. Après l’intégration de l’algorithme d’évitement de collision à la méthode FABRIK figure 3.6-c un résultat satisfaisant est obtenu où l’extrémité atteint la cible au même temps que les jointures évitent les obstacles.

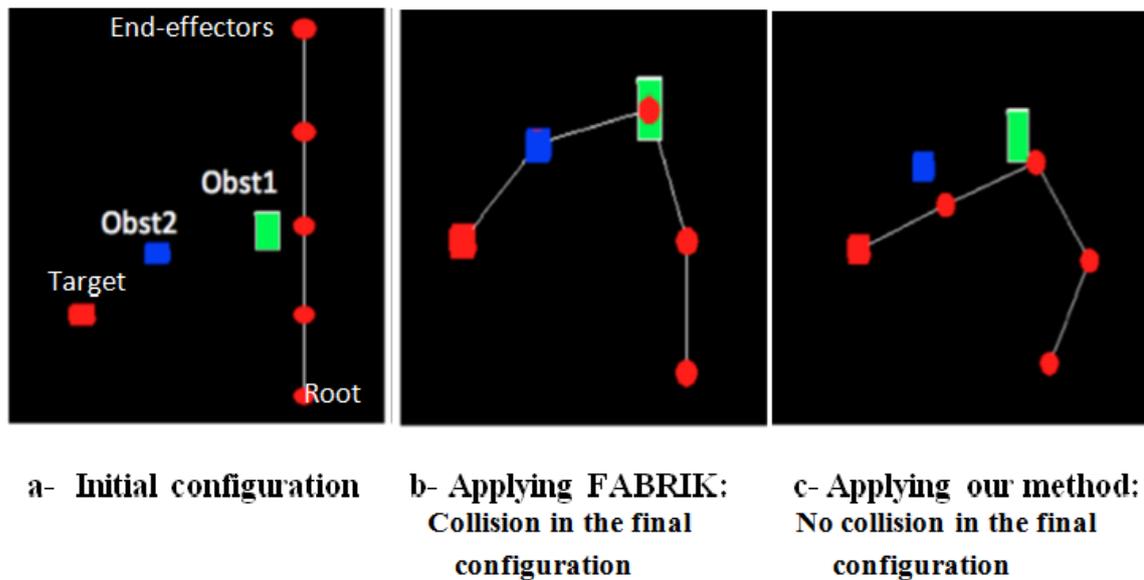


FIGURE 3.6 – Évitement d'obstacles

3.4.2 Comparaison avec PIK

La méthode PIK proposé par Baerlocher1 [9] gère le problème de conflit entre les tâches en proposant de distribuer des priorités à celles ci. La méthode est basée sur la matrice Jacobienne ce qu'implique qu'elle souffre de problème de singularité. Un autre limite de cette méthode a été présenté par Andreas [5], qui a comparé la méthode FABRIK avec la méthode Jacobienne, il a montré que FABRIK est plus rapide. D'après nos tests, lors de l'exécution de multiple tâches en même temps, notre méthode à base de priorité donne des résultats plus rapides que la méthode FABRIK, c'est pourquoi on peut déduire que notre méthode à base de priorité est plus rapide que PIK qui se base sur la méthode Jacobienne.

Dans le tableau 3.4.2, nous représentons certains critères de similarité entre notre travail et ceux des autres travaux que nous avons nous même sélectionnés. Le tableau résume les principaux critères de chaque méthode de résolution. À la fin de ce tableau, nous avons inséré notre travail afin de l'évaluer sur les mêmes critères et de pouvoir ainsi présenter ses points forts, ainsi que les points en commun. Les méthodes à base d'exemple représenté dans le tableau par le travail de Grochow [37] ne prends pas en compte les collision entre les jointures tout comme la méthode FABRIK [5]. Tandis que la méthode SIK proposé pr Unzueta [103] prends en compte uniquement les autos collisions.

Notre méthode PIKOV [12] repose sur le principe d'algèbre géométrique ce que rend

l'obtention des résultats très rapide comme dans la méthode FABRIK. Mais elle diffère de cette dernière, par la prise en compte la présence d'obstacles lors du calcul des nouvelles positions de jointures, en plus elle peut contrôler plusieurs jointures en même temps. Un autre point avantageux consiste à l'obtention des résultats sans le besoin d'une base de mouvement.

Méthode	Principe	Nombre d'effecteur	Obstacle
SIK [103]	Itérative, Analytique	Multiple	Seulement les auto-collisions
PIK [10]	Optimisation	Multiple	–
Style based IK [37]	Apprentissage	1 seul	–
FABRIK [5]	Algèbre géométrique	Multiple avec conflit	–
PIKOV [12]	Algèbre géométrique	Multiple	Oui

TABLE 3.1 – Comparaison différentes méthodes de cinématique inverse

3.5 Conclusion et discussion

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 ces derniers des méthodes reposent sur la cinématique inverse. La plupart des méthodes de IK souffrent : de temps de calcul, de la complexité et du manque de réalisme. Dans ce chapitre, une nouvelle méthode est proposée en se basant sur le principe d'une méthode à base d'algèbre géométrique FABRIK afin de surmonter les problèmes des méthodes existantes et pour répondre aux exigences de l'application où nous allons utiliser la méthode proposée..

Notre méthode permet de contrôler des postures des personnages d'une manière rapide et réaliste due à ces fonctionnalités. Elle permet d'atteindre plusieurs objectifs simultanément d'une manière rapide et sans conflit en introduisant le principe de priorité. En plus, notre méthode prend en compte l'existence des obstacles dans la scène.

Les résultats obtenus sont encourageants et permettent d'optimiser le temps de calcul et d'éviter des cas de conflits. Deux techniques à base de priorité sont proposées et testées la pondération des et l'utilisation de priorités. Dans la première technique, le choix de la valeur de priorité est très important, un mauvais choix peut alourdir l'application. Alors que dans la deuxième, la tâche la plus prioritaire est satisfaite et les résultats ne dépendent que de la tâche la plus prioritaire, le problème de singularité est aussi évité en introduisant

le principe de priorité.

La méthode traite aussi le problème de la cinématique lors de l'existence des obstacles dans l'environnement. L'application de l'algorithme d'évitement d'obstacle après l'application de la méthode de la cinématique inverse peut nuire aux contraintes vérifiées par celle-ci, pour cette raison, nous avons intégré l'algorithme d'évitement d'obstacle dans le processus de la cinématique inverse.

Les méthodes de la cinématique inverse existantes, traitent un seul problème à la fois, soit elles traitent le problème de conflit entre les tâches soit l'évitement d'obstacle, à notre connaissance, il n'existe pas une méthode qui traite : le problème d'évitement d'obstacle, de conflit entre les tâches tout en gardant la rapidité d'obtention des résultats en même temps.

Nous utiliserons cette méthode dans un système complet d'édition de mouvements développé dans le cadre de notre projet. Dans le prochain chapitre, nous présenterons notre méthode de modification de mouvements où nous intégrons la méthode de la cinématique inverse présentée dans ce chapitre.

PIKOV pour la Modification de trajectoire

4.1 Introduction

Dernièrement, les méthodes d'animation traditionnelles sont remplacées par des méthodes d'animation à base de mouvements déjà capturés. Les systèmes de captures de mouvement sont devenus les méthodes les plus utilisées dans l'animation. 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.

Donc, le mouvement capturé a besoin d'être modifié afin de satisfaire les besoins de l'utilisateur. Cependant, les systèmes récents de modification se concentrent sur la synthèse des mouvement en utilisant une base de mouvements. Néanmoins, on ne peut pas synthétiser tous les styles en utilisant ce type de méthodes, il manque aussi d'une interface interactive pour que l'utilisateur contrôle le mouvement. Par conséquent, il est difficile d'exprimer les intentions exactes des utilisateurs, nous cherchons donc à introduire une technique facile et interactive pour la modifications des mouvements induisant des résultats naturels. La technique proposée permet aux utilisateurs d'exprimer leur besoins, ainsi que, 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 dans le but consiste à modifier une posture définissant des contraintes cinématiques. La figure 4.1 représente une description de la procédure de modification d'un mouvement.

4.2 Aperçu général du système d'édition des mouvements

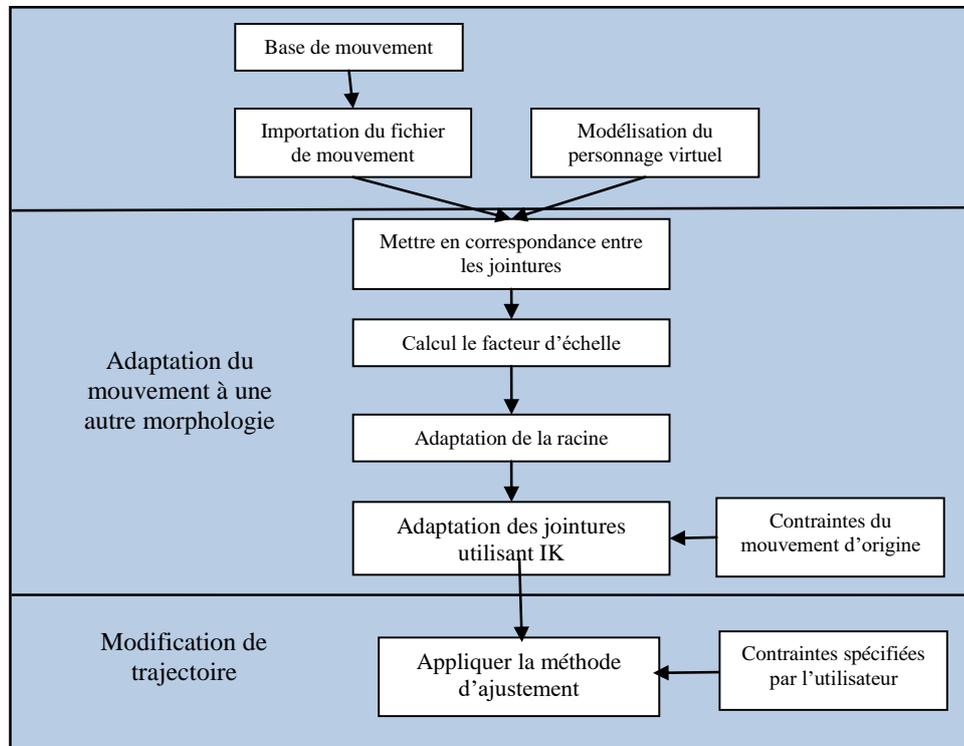


FIGURE 4.1 – Vue globale du système proposé d'édition des mouvements

Le système est constitué en trois modules : le premier sert réutiliser un mouvement depuis une base tel qu'il est sans modification sur un personnage virtuel déjà modélisé. L'adaptation de mouvement à une autre morphologie ainsi que la modification de trajectoire ont pris une place au sein de notre système.

4.2.1 Adaptation du mouvement à une autre morphologie

C'est un module très essentiel dans notre système d'édition du mouvements. Lors de la lecture du fichier, le sujet peut n'avoir pas les mêmes caractéristiques que notre personnage déjà défini. Nous avons proposé une méthode afin de gérer ce problème. L'explication détaillée est donnée dans le chapitre 5.

4.2.2 Modification de trajectoire

L'objectif de ce module est de modifier la trajectoire d'une ou des jointures en spécifiant des contraintes à respecter. Pour atteindre cet objectif nous avons proposé une méthode pour la modification de trajectoire des jointures d'une façon interactive.

Après cette présentation globale du système d'édition des mouvements, nous nous concentrons sur le module de la modification de trajectoire dans le reste du chapitre. Dans les sections suivantes, nous décrirons en détail la méthode proposée. Enfin, nous donnerons quelques résultats, et une comparaisons avec d'autres méthodes qui ont le même objectif.

4.3 Modification de trajectoire

L'objectif de ces méthodes est de modifier un mouvement d'origine dont l'utilisateur ajuste la trajectoire du mouvement pour chaque jointure afin de créer un nouveau mouvement. Afin d'atteindre ce but, des chercheurs ont utilisé les méthodes à base de traitement de signal, d'autres ont utilisé des méthodes à base de contraintes. Nous notons ici que, la modification de trajectoire peut être interactive ou autonome, cela dépend des besoins de l'application.

4.3.1 Interactivité

La possibilité de modifier un mouvement d'une manière interactive est essentielle pour l'objectif de notre projet. On présente une nouvelle technique qui permet aux utilisateurs de modifier la trajectoire des jointures d'un personnage virtuel en utilisant la souris et le clavier d'une manière simple et pour les utilisateurs non professionnels.

Généralement, le but des systèmes d'édition de mouvement est d'apporter les modifications désirées sur un mouvement existant en préservant la qualité et les caractéristiques essentielles du mouvement original autant que possible. C'est une façon rapide de réutiliser les mouvements qui sont utiles pour les animateurs novices.

Nous cherchons dans notre travail à proposer une stratégie générale pour modifier n'importe quel mouvement d'une façon interactive et qui peut être utilisée par un utilisateur non expert. Aussi, les résultats de la méthode proposée ne doivent pas être liés à un type ou une structure d'un personnage bien définie.

4.3.2 Synergies

La synergie (selon Larousse) n.f. (gr. sunergia, coopération). 1. physiол. Association de plusieurs organes pour l'accomplissement d'une fonction. 2. Mise en commun de plusieurs actions concourant à un effet unique avec une économie de moyens.

Au niveau du mouvement des personnes, il existe plusieurs corrélations entre les jointures durant un mouvement. Cependant, les jointures ne sont pas corrélées de la même manière. Par exemple, durant une séquence de marche, on peut trouver une corrélation entre la jambe et le genou mais pas entre la jambe et la tête. En général, les jointures de la même sous chaîne cinématique seront corrélées l'une à l'autre parce qu'elles fonctionnent ensemble pour une tâche particulière. Une bonne méthode de modification du mouvement doit prendre cet aspect en considération afin d'obtenir des mouvements plus naturels. Par exemple, Kulpa dans son travail [62] a divisé le personnage virtuel (personnage articulé) en des groupes « sub-structures » indépendantes. Cependant, l'inconvénient majeur est le manque de synergie globale entre les groupes, ce qui peut donner des mouvements irréalistes.

4.4 État de l'art sur les méthodes interactive d'édition de mouvement

Dans le premier chapitre, nous avons déjà présenté un état de l'art sur les différentes méthodes et travaux dans le domaine de l'édition du mouvement. Parmi ces travaux, il existe ceux qui sont centrés sur la modification de trajectoire d'une façon interactive.

4.4. ÉTAT DE L'ART SUR LES MÉTHODES INTERACTIVE D'ÉDITION DE MOUVEMENT

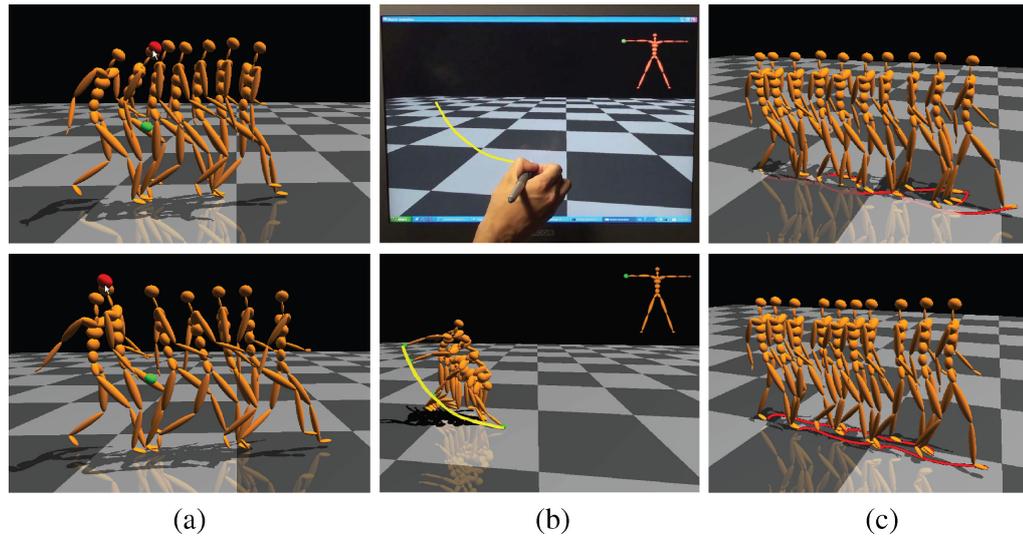


FIGURE 4.2 – Génération des mouvements avec un modèle déformable de mouvement : a. Interface de manipulation par glissement d'un point (en rouge); b. Interface à base de croquis par stylo; c. filtrage de mouvement et élimination de patinage des pieds

Lecalenc [10] a utilisé une méthode à base de la cinématique inverse PIK afin de modifier un mouvement d'une manière interactive. Il a spécifié deux types d'interactivités, soit l'utilisateur spécifie la position finale de l'extrémité -ou n'importe qu'elle jointure- ou il spécifie une trajectoire à suivre par une jointure. Sa méthode donne de bons résultats, mais comme elle est basée sur la méthode jacobienne, elle souffre de problème de la singularité. Jianyuan Min [74] a utilisé une méthode statistiques afin de créer des mouvements réalistes qui satisfont différentes formes de contraintes définies par l'utilisateur. Malgré que cette méthode donne des postures réalistes, elle ne donne pas des résultats lors des conflits entre les tâches exécutées simultanément. En plus, le système ne génère pas des mouvements dans le cas d'absence des caractéristiques souhaitées dans la base. Par exemple, le modèle déformable de la marche ne peut pas générer une marche en agitant la main parce que la base de mouvement ne contient pas cette caractéristique "main agité". Michael Neff [82] a proposé un système d'édition du mouvement dont le but est de modifier le style de mouvement introduisant les corrélations entre les jointures lors du mouvement.

4.4. ÉTAT DE L'ART SUR LES MÉTHODES INTERACTIVE D'ÉDITION DE MOUVEMENT

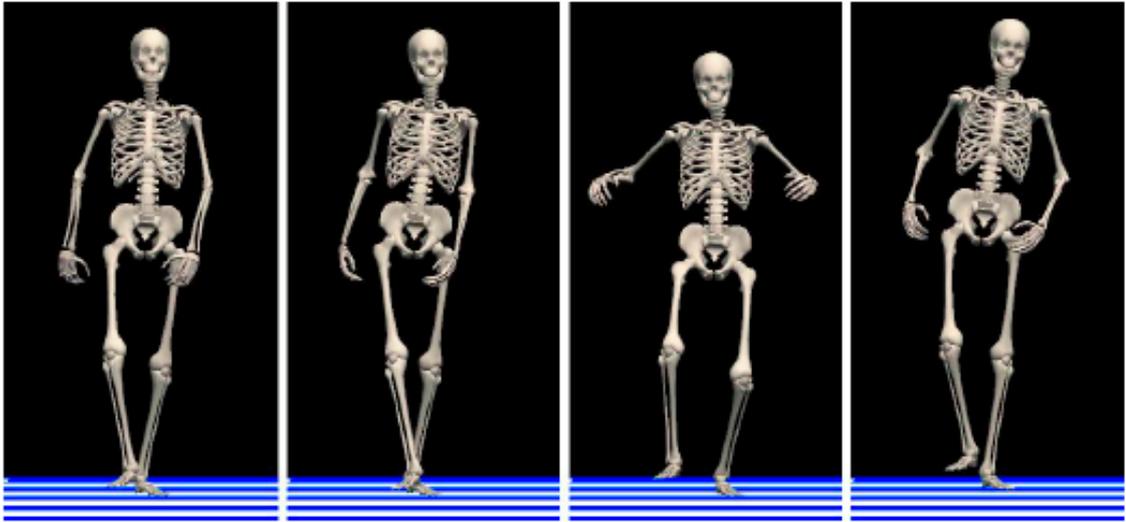


FIGURE 4.3 – Un mouvement de la marche (à gauche) An original walking motion (left) is edited to produce a runway walk, a sneaky walk and a limp. est édité pour créer un

Xiaolin K [108] a proposé une méthode interactive basée sur une très grande base de mouvement afin de poser les jointures du personnage virtuel. Il a aussi développé une interface pour effectuer la modification par un utilisateur non professionnel. En plus, Ikemoto et al. [46] ont proposé une technique d'édition de mouvement à base de processus gaussien et une méthode de cinématique inverse simple. Le système est capable de propager les modifications d'une posture à travers la séquence du mouvement.

4.5 Proposition

Travail	Principe	Caractéristiques
Benoit [65]	Cinématique inverse PIK	<ul style="list-style-type: none"> – Méthode per-frame – Interactive
Kulpa [62]	Cinématique inverse CCD	<ul style="list-style-type: none"> – Subdivision du personnage virtuel – Manque de synergie entre les groupes
Jianyuan Min [74]	Méthode statistique	<ul style="list-style-type: none"> – Satisfaire plusieurs contraintes – Posture réaliste – Base de mouvement large – Hors ligne – Génération des nouveaux mouvements
Michael Neff [82]	Cinématique inverse à base d'exemple	<ul style="list-style-type: none"> – Changement de Style – Méthode per-frame
Xiaolin [108]	Méthode à base d'exemples	<ul style="list-style-type: none"> – Interactive – Temps réel – Base de mouvement large

TABLE 4.1 – Comparaison entre des méthodes de modification des mouvements

Le tableau 4.5 regroupe des différents travaux sur la modification des mouvements avec leurs caractéristiques. L'objectif de notre travail est de s'éloigner des méthodes de modification récentes qui sont basées sur l'utilisation d'une base de mouvement. Outre la nécessité d'une large base de mouvements afin de les utiliser, ce qui implique un grand espace de stockage. Aussi, l'absence des résultats lorsque les contraintes à vérifier n'existent pas dans la base. Notre contribution, consiste à proposer une méthode de modification qui sert à ajuster le mouvement selon les contraintes proposées par l'utilisateur en :

4.5. PROPOSITION

- subdivisant le corps du personnage virtuel en groupes.
- appliquant la méthode de la cinématique inverse PIKOV pour chaque groupe, ce qui permet :
 - d'utiliser la stratégie de priorité des tâches afin d'éviter le conflit entre les tâches simultanées.
 - d'intégrer un algorithme d'évitement d'obstacles
- Impliquant des corrélations entre les parties du corps.

Nous avons proposé une méthode à base de la cinématique inverse afin d'ajuster une posture de tout le corps du personnage virtuel définissant des contraintes cinématiques. La figure 4.4 représente une description de la procédure de modification d'un mouvement.

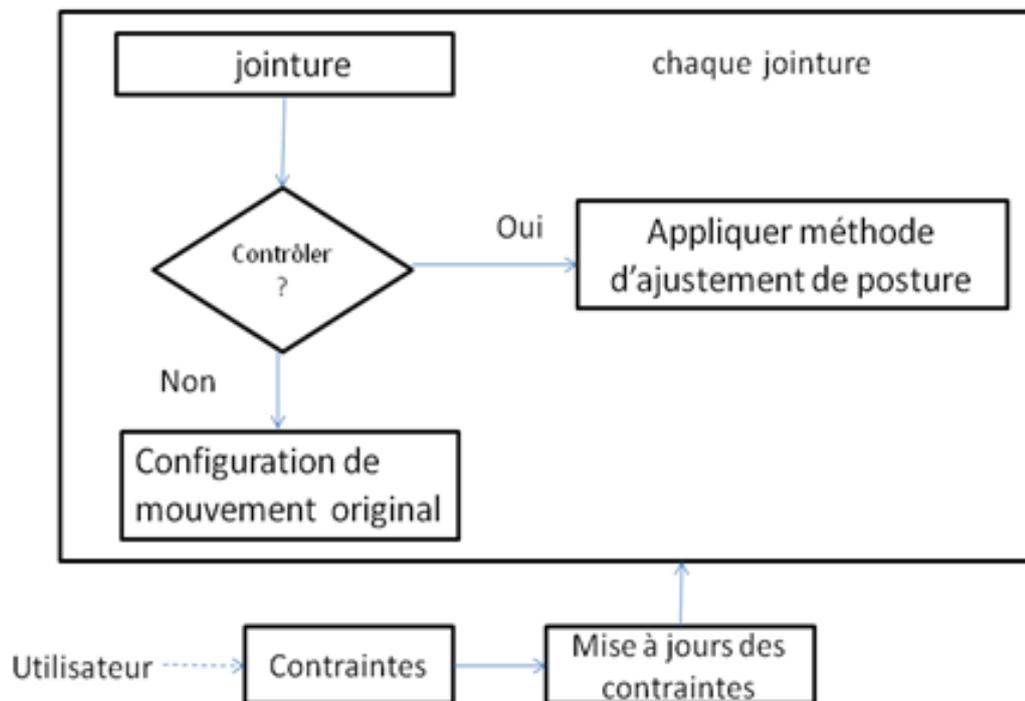


FIGURE 4.4 – Processus d'ajustement d'une posture

Notre algorithme proposé, est une méthode *per-frame based motion editing* à base des contraintes où on peut ajuster chaque frame interactivement. Donc, l'orientation de chaque jointure est définie selon la présence des contraintes ou pas. Dans le cas où elle est contrôlée, on applique la méthode d'ajustement de posture que nous avons proposée et qui est basée sur la cinématique inverse afin qu'elle réponde aux nouvelles contraintes. Dans le cas contraire, on applique juste les mêmes angles de jointure depuis le fichier de mouvement.

La modification des mouvements d'une manière interactive à besoin d'une méthode exécuté à chaque frame *per-frame* ce qui donne la possibilité de modifier les postures à chaque pas de temps. Pour ces raisons, nous avons choisi une méthode de modification *per-frame*.

4.6 Aperçu de la méthode proposée

La figure 4.5 représente un aperçu général de la méthode proposée où le personnage est divisé en cinq groupes qui représentent : les jambes, le tronc, le bras et la tête. Les groupes sont connectés entre eux par deux jointures, la racine et la « sterno-clavicule », nous avons choisi des connexions entre les groupes afin d'assurer la corrélation et la synergie entre les parties du corps du personnage. Plus de détails sur l'étape de subdivision seront donnés dans la section qui traite du modèle du corps du personnage. La subdivision du corps a l'avantage de faciliter l'ajustement, alors que la création de synergie entre les parties du corps assure le réalisme des résultats de l'étape d'ajustement.

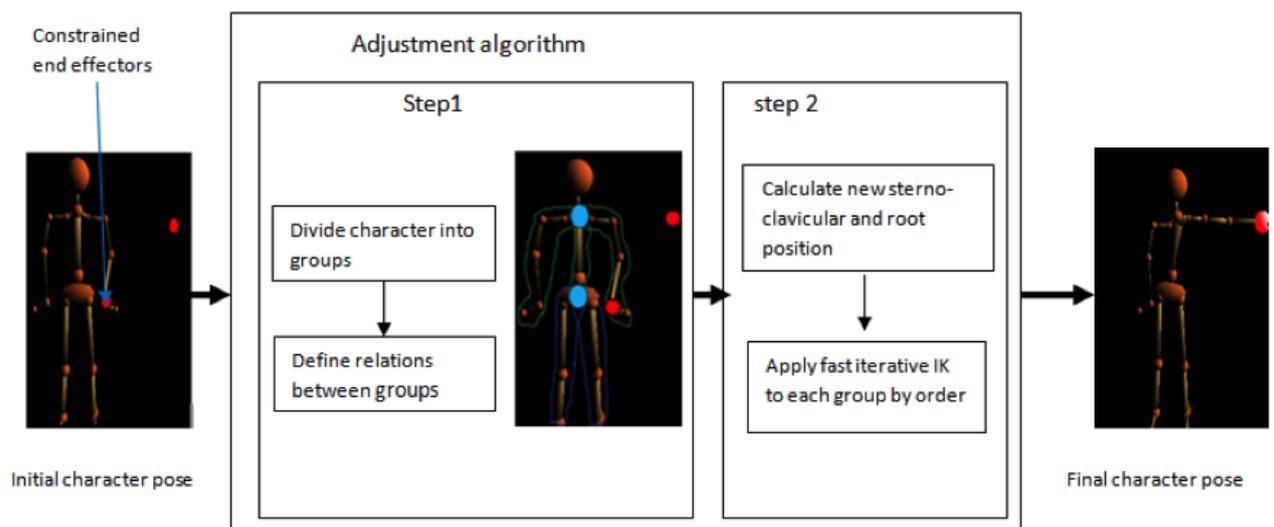


FIGURE 4.5 – System overview

Afin d'ajuster la posture à de nouvelles contraintes utilisant notre méthode, nous devons avoir la posture initiale du personnage comme entrée, les jointures à contraindre et leur nouvelles positions souhaitées. Comme une première étape, la configuration de racine (positions/orientations) sont calculées en se basant sur les positions désirées des jointures. Ensuite, la méthode de la cinématique inverse PIKOV est appliquée sur chaque groupe par ordre afin de respecter les contraintes et d'avoir une posture réaliste. A chaque itération

de la cinématique inverse ont doit définir les limites dans le cas où cela est nécessaire, afin de respecter les limites biomécaniques pour chaque jointures, nous utiliserons le même principe [5].

4.7 Contraintes

Notre travail se situé dans la classe des méthodes d'édition de mouvements à base de contraintes. Dans cette classe de méthodes, l'utilisateur à besoin de spécifier des contraintes afin de modifier un mouvement interactivement ou même pour adapter le mouvement à une autre morphologie. ces derniers, sont de nature géométrique : position et orientation. En plus, elles sont intuitives à définir même par un utilisateur qui n'est pas expert.

Une contrainte a besoin d'un ensemble de paramètres, nous signalons qu'il existe des paramètres communs entre toutes les contraintes comme la durée d'activation, l'instant d'activation, et d'autres paramètres dépendent du type de contraintes : par exemple pour atteindre un but, on a besoin de paramètres de la position de but, alors que pour l'adaptation de mouvement à une autre morphologie, on a besoin des fois de conserver une orientation. Enfin, il existe une contrainte qui doit être toujours vérifiée "les contraintes de support". Ces contraintes sont utilisées afin d'assurer le respect des appuis des pieds du personnage sur le sol : bipodale(les deux pieds au sol). Monopodale(un seul pied au sol) ou aucun appui (phase aérienne). Le type de support est défini généralement par l'utilisateur.

Dans notre recherche, nous souhaitons pouvoir contraindre n'importe quelle jointure du corps du personnage afin de permettre aux utilisateurs de modifier facilement et rapidement un mouvement qui existe déjà. En particulier, les contraintes géométriques, comme la position de l'extrémité dans l'espace 3D ou la trajectoire d'une extrémité, sont plus intuitives pour la manipulation interactive, parce que l'utilisateur peut spécifier le but juste par glissement de l'extrémité à la nouvelle position. Nous équipons l'animateur avec des contraintes d'image clé comme le montre la figure 4.6

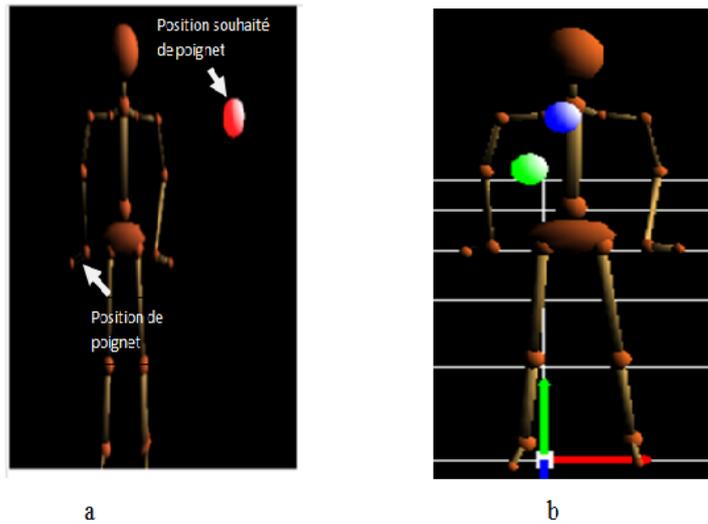


FIGURE 4.6 – Exemple de contraintes spécifiées par l'utilisateur : a- une seule contrainte, b- deux contraintes

4.7.1 Contraintes des images clés

Dans ce type de contraintes, l'extrémité est attachée directement au corps de l'humanoïde virtuel ensuite elle est glissée « dragged » à sa nouvelle position. Key-frame contraintes sont des contraintes de positions, c'est à dire, l'extrémité ne suit pas une trajectoire spécifique. La Figure 4.6 décrit ce type de contraintes. Une contrainte d'image clé sont représenté simplement avec un point 3D.

4.7.2 Contraintes de trajectoire clé

Elles permettent aux animateurs de modifier un mouvement, en spécifiant une trajectoire à suivre par une extrémité. L'objectif de cette formulation est de réduire le travail de l'animateur qui sert à spécifier manuellement toute la trajectoire de l'extrémité pour tout le mouvement. Par exemple, supposons que l'animateur veut modifier un mouvement de préhension, alors, au lieu de spécifier un ensemble de points de trajectoire de la main, qui peut être encombrant, il peut définir juste les positions importantes que la main doit passer à travers, ensuite toute sa trajectoire est générée automatiquement. Afin de fournir cette fonctionnalité, nous utilisons une méthode d'interpolation pour calculer l'ensemble des points 3D des contraintes. Le système génère automatiquement la trajectoire complète de l'extrémité.

4.8 Ajustement utilisant la cinématique inverse PIKOV

Cette section présente la structure de personnage virtuel ceux qui est utilisé par nos trois modules de système d'édition de mouvement 4.1. Puis, nous détaillons la méthode d'ajustement proposée.

4.8.1 Modèle du personnage

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.

Par exemple, Wang [106] et Karabinova Makki [53] se sont intéressés respectivement aux membres supérieurs et aux membres inférieurs. Les modèles proposés par ces auteurs sont composés de trois segments (bras/cuisse, avant-bras/jambe, main/pied) avec 7 degrés de liberté (3 ddl pour l'épaule et la hanche, 2 ddl pour le coude et le genou, 2 ddl pour le poignet et la cheville). Fradet [32] pour la simulation du tir u handball a proposé un modèle composé de 14 segments à 31 degrés de liberté. Pour la modélisation dynamique de la marche et l'évitement d'obstacles, el Hafi [29]a développé un modèle avec 11 segments et 16 ddl). Le tronc est, dans cette étude, représenté en une seule partie.

Pour la modélisation des différents type de mouvement tel que le cas de notre étude, nous allons essayer de concentrer sur les deux parties du corps du personnage virtuel, la partie haute et la partie inférieure tout en considérant les articulations les plus importantes (hautes et basses).

Notre personnage virtuel est constitué de 22 articulations,20 segments et avec 31 DDL. La figure 4.7 montre l'ensemble des jointures de personnage virtuel avec la description des relations père/fils entre les jointures.

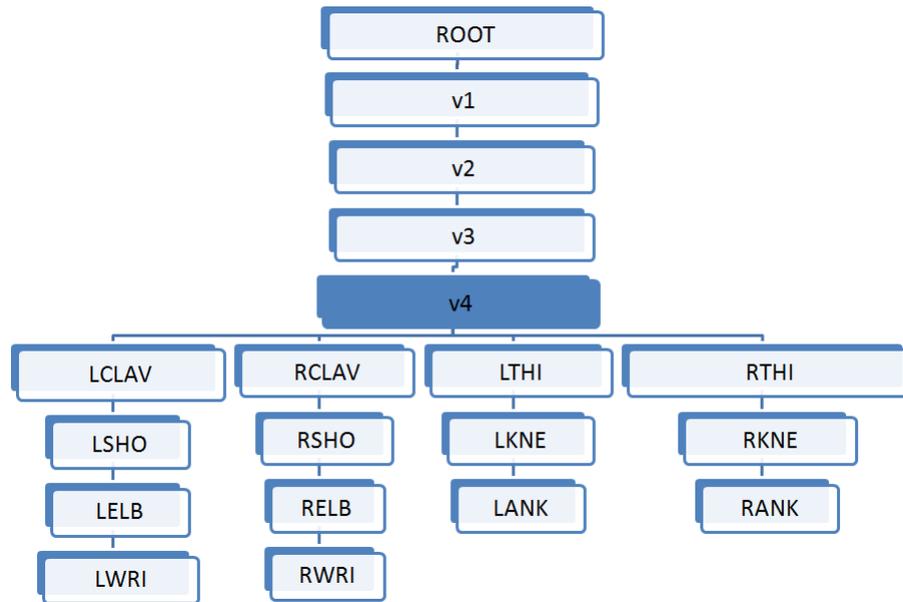


FIGURE 4.7 – Hiérarchie de personnage virtuel

La structure de notre personnage consiste en des chaînes cinématiques rigide (cylindres » connectées entre eux par des jointures angulaires. Une représentation simplifié est donnée dans la figure 4.8

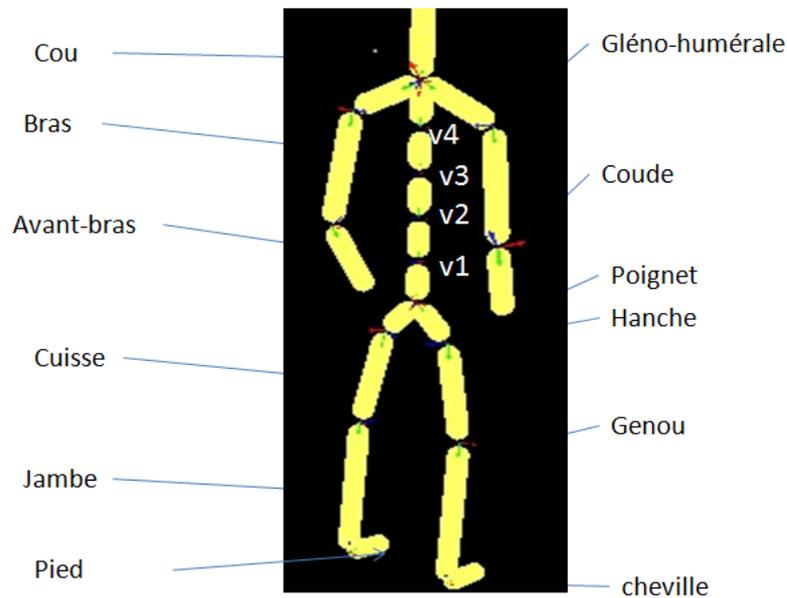


FIGURE 4.8 – Modèle du personnage : hiérarchie proposée

Les degrés de liberté sont réparties comme suit : le pelvis a 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 la gléno-humérale ont trois degrés de liberté. figures 4.9,

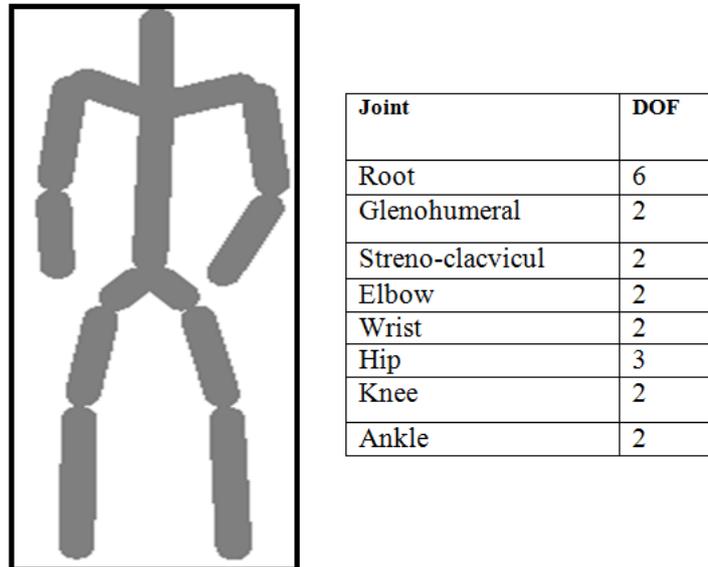


FIGURE 4.9 – Modèle du personnage : Degré de liberté

Le squelette du personnage virtuel est composé de plusieurs chaines cinématiques. Ces chaines partent de la racine du personnage qui est le pelvis et vont jusqu'aux extrémités : les pieds et les poignets (section 1.2.1). Cette représentation pose le problème de partage du tronc par plusieurs chaines cinématiques. Ainsi, lorsque deux contraintes sont placé sur chaque bras, chaque contrainte apporte des modifications sur le tronc, ce qui peut conduire à des modifications adverse de ces segments.

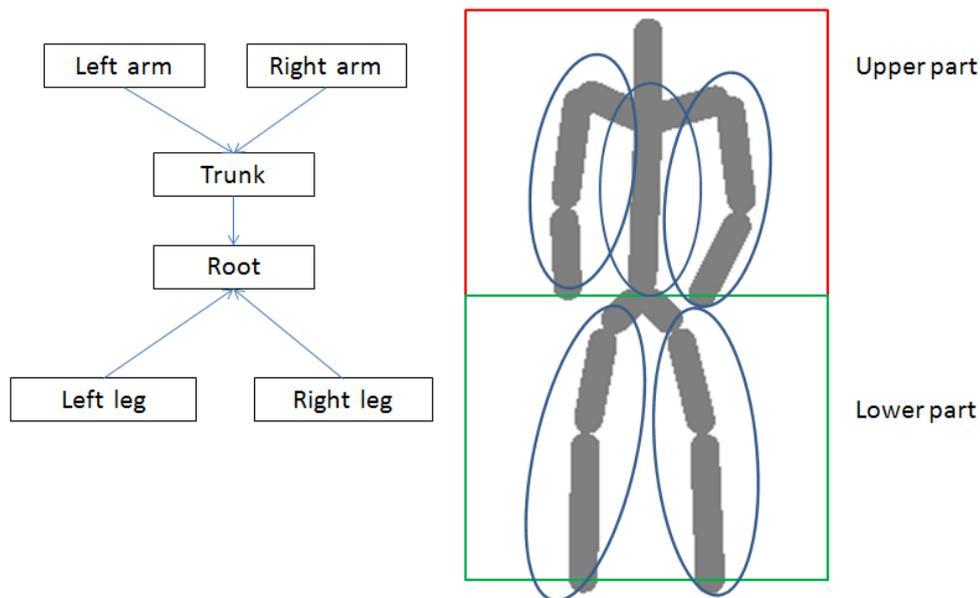


FIGURE 4.10 – Représentation du personnage : découpage de personnage en groupe

Dans le travail de Shin [93] le personnage virtuel est subdivisé en groupe ensuite une méthode de la cinématique inverse analytique est appliquée à chaque groupe, nous avons suivis le même principe dans notre travail où le personnage en deux parties et cinq groupes (figure 4.10).

Nous proposons donc de découper le squelette en deux parties et cinq groupes qui définissent les chaînes cinématiques. Notre découpage est présenté dans le figure 4.10 permet de séparer partiellement les chaînes cinématiques. Contrairement au travail de Kulpa [62] qui a décomposé le corps du personnage en des groupes séparés totalement, nous avons garder des points de contact entre les différentes parties du corps du personnage virtuel.

Les bras et les segments de tronc sont liés par la « stéréno-clavicule », où le changement dans la configuration d'un bras affecte le changement dans l'autre bras, elle peut même affecter des modifications sur la configuration de tronc. En plus, les jambes et le tronc sont liées par la « racine ». Le changement dans la configuration d'une jambe peut impliquer le changement dans la racine et par la suite le changement dans l'autre jambe. Pour mieux expliquer cette corrélation, supposant que nous voulons bouger le pied gauche afin d'être dans une position définie, alors que cette nouvelle position est loin de la jambe, afin d'atteindre cet objectif nous devons changer la position de la racine, ce changement affecte les autres membres comme l'autre jambe.

Re positionner ou orienter la racine implique le changement dans les configurations de tous les groupes, cette représentation permet de faciliter le contrôle du personnage

virtuel. Après la subdivision, on applique la cinématique inverse pour les groupes concernés ce qui rend l'ajustement plus rapide. Cette performance est due à l'application de la méthode seulement sur les jointures des groupes concernés par le mouvement. Notre méthode permet également de spécifier l'ordre dans lequel l'ajustement est effectué. En restant conforme à la hiérarchie du squelette qui part de la racine du personnage vers les extrémités, nous avons défini un ordre équivalent pour les groupes, parmi lesquels, nous avons ajusté la racine du personnage en premier, puis en suivant le lien de l'hierarchie, les jambes et le tronc et enfin on fait l'ajustement des bras et de la tête.

4.8.2 Ajustement du corps du personnage

L'idée générale présentée dans cette thèse pour ajuster un mouvement consiste en la résolution de l'ajustement d'une manière séquentielle, en appliquant une méthode itérative de la cinématique inverse déjà présentée dans le chapitre précédent pour les différentes parties dans un ordre bien définie. La méthode proposée n'utilise pas une base de données, ce qui donne l'avantage de l'absence des problèmes liées à l'utilisation de la base, comme l'espace de stockage et l'étape de pré-traitement. Aussi, Nous pouvons contraindre chaque point, les exemples données dans la thèse seront sur les extrémités mais la méthode peut contraindre tout type de jointure.

En se basant sur les positions souhaitées des extrémités, la configuration de la racine et celle de la stéréno-clavicule sont calculées. Les configurations des jointures du tronc sont calculées en se basant sur les nouvelles configurations de la racine et de la stéréno-clavicule. Finalement les nouvelles configurations de chaque membre sont déterminées en utilisant la méthode de la cinématique inverse tenant en compte la nouvelle position de stéréno-clavicule pour les bras, et sur la configuration de racine pour les jambes. L'algorithme 1 résume la méthode de l'ajustement proposé.

Algorithme 2 Algorithme d'ajustement de corps du personnage

- 1: Entrée : posture initiale du personnage, Position souhaitée des jointures
 - 2: Sortie : Posture finale du personnage
 - 3: Calcul les nouvelles positions de la racine et la stéréno-clavicule en se basant sur les positions souhaitées des jointures
 - 4: Ajustement de tronc, en se basant sur la position de stéréno-clavicule
 - 5: Ajustement des bras
 - 6: Ajustement des jambes
 - 7: Fin
-

4.8.2.1 Configuration Streno-clavicule et racine

Notre algorithme prend en compte les règles biomécaniques d'organisation ségmentaire : où on ne sollicite le tronc que si les bras ne sont pas suffisants pour atteindre l'objet. Pour avoir une posture naturelle, notre proposition s'inspire de la nature, nous vérifions premièrement la contribution du tronc et de la racine dans le mouvement des bras ou des jambes afin de satisfaire les contraintes, puis nous calculons la nouvelle configuration du tronc.

Pour tester la contribution du tronc dans le mouvement, on calcule la distance D entre stéréno clavicule et la cible, si elle est inférieure à la longueur du bras, la stereno clavicule et le racine gardent leurs positions d'origine (figure 4.11) sinon le tronc change de configuration (figure 4.12).

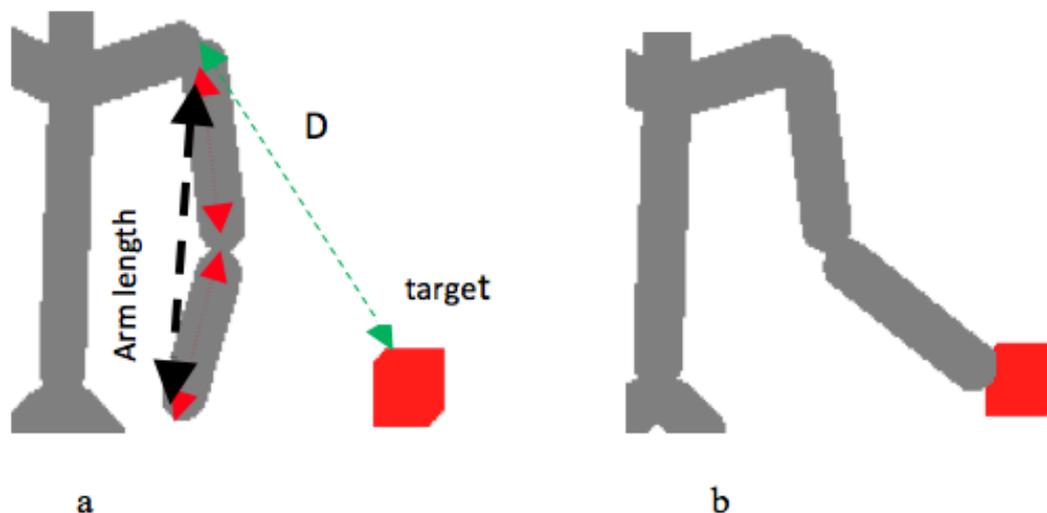


FIGURE 4.11 – Préhension d'un cible avec une chaîne simple. Figure *a* représente la configuration initiale du bras où nous calculons la longueur du bras et la distance entre la cible et l'extrémité. Dans la figure *b* nous déplaçons seulement le bras parce que sa longueur est inférieure à la distance entre la cible et les extrémités

Dans le cas où la cible est placée au delà de la longueur du bras, les segments du tronc sont intégrés dans le mouvement du bras. Dans ce cas, la nouvelle position/ orientation de stereno clavicule est calculée afin d'obtenir une distance entre la cible et la position calculée égale à la longueur du bras.

S'il existe plus d'une seule contrainte à atteindre, l'utilisateur peut choisir entre deux possibilités déjà discutées dans la section 3.3.2 afin de calculer la nouvelle position de stereno-clavicule et de la racine.

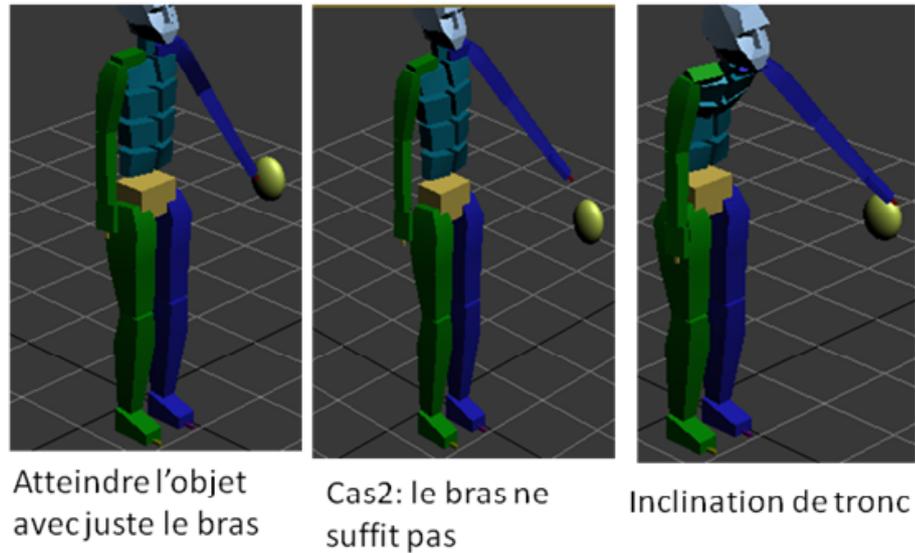


FIGURE 4.12 – Cas d'inclination du tronc dans le mouvement

Dans certaines situations, nous ne pouvons pas déplacer le tronc afin que le stereno clavicule soit dans sa nouvelle position pos' -la cible est plus loin-. Une solution sert à déplacer la racine avec une erreur de distance pour atteindre pos'_{pelvis} . Cette erreur est calculée en utilisant la distance entre la position actuelle de pelvis et la position actuelle de la stereno clavicule, et la distance entre la position de stereno clavicule et Pos' .

Dans le cas où il y a deux tâches à atteindre, et afin de calculer (par exemple lorsqu'on pose des contraintes sur les deux bras), la nouvelle configuration du tronc est calculée en considérant la stereno-clavicule comme sub-joint entre les deux bras. La nouvelle position de stereno-clavicule est calculée en choisissant une stratégie parmi les stratégies déjà proposées dans la section 3.3.2. Si l'utilisateur n'a choisi aucune stratégie, nous calculons cette nouvelle position en utilisant la stratégie de pondération (section 3.3.2.2).

Lors de l'existence des contraintes sur la partie supérieure et inférieure du personnage, et les contraintes des parties inférieures concernent l'appuis des pieds sur le sol, dans ce cas la nouvelle position de la racine est calculée par défaut en utilisant la stratégie de priorité (section 3.3.2.1) de où les contraintes de la partie inférieure ont la plus haut priorité.

4.8.2.2 Ajustement de la racine

Généralement, on impose que la racine ne change pas de configuration lors de la modification du mouvement. Néanmoins, il existe des situations où il est préférable de changer un peu la configuration de la racine afin d'effectuer une tâche. Cette modification

doit tenir en compte la critère d'appuis au sol.

La détermination de la position de la racine est basée sur la position de la stérenoclavicule et la position désirée du pied. Les contraintes sur les pieds ont une grande priorité. La nouvelle position de la racine est calculée utilisant la stratégie présenté dans la section (section 3.3.2.1). Nous avons choisi cette stratégie parce que la nouvelle position de racine doit vérifier impérativement les positions des pieds sur le sol sinon le personnage tombe.

Une fois le calcul des configurations des jointures de relation, qui relie entre tous les membres du corps du personnage est fait, l'étape suivante, consiste à modifier et à ajuster les configurations des membres du corps du personnage suivant l'ordre hiérarchique.

4.8.2.3 Ajustement du tronc

L'objectif de cette étape est de trouver l'orientation(/ la position) des jointures du tronc en se basant sur la configuration désirée de la stéreno-clavicule. Cette étape est appliquée généralement lorsqu'on ne peut pas atteindre un objet seulement avec le bras.

Le tronc est constitué de cinq jointures qui représentent la colonne vertébrale. Afin d'ajuster la configuration pour atteindre un objectif, nous appliquons la cinématique inverse, pour cela nous avons besoin de spécifier la racine et l'extrémité de la chaîne où nous allons l'appliquer. Dans ce cas, la racine de la chaîne est la racine du personnage virtuel "pelvis" et son extrémité est la stéreno-clavicule.

L'algorithme 3 résume cette étape.

Algorithme 3 Algorithme 1 d'ajustement du tronc

- 1: Entrée : Position du cible, configuration initiale du tronc
 - 2: Sortie : Configuration du tronc
 - 3: $D1 = | \text{Position de la stéreno-clavicule} - \text{Position du poignet} |$
 - 4: $D = | \text{Position de la stéreno-clavicule} - \text{Position du cible} |$
 - 5: if($D1 < D$) Calculer la nouvelle position du stéreno-clavicule
 - 6: appliquer PIKOV afin de trouver la nouvelle configuration du tronc
 - 7: Fin
-

Dans le cas de l'application des contraintes sur les deux bras, la nouvelle configuration du tronc est calculée en tenant compte que la stéreno clavicule est la jointure qui relie les deux bras. La nouvelle position de stéreno clavicule est calculée en choisissant une stratégie parmi les stratégies décrites dans le chapitre précédent. Par défaut, si aucune stratégie n'est choisie, cette nouvelle position est calculée utilisant l'équation définie dans la section 3.3.2.2. Une fois que nous avons calculé la position, nous appliquons la cinématique inverse au segment du tronc pour obtenir la nouvelle configuration du tronc.

Algorithme 4 Algorithme2 d'ajustement du tronc

- 1: Entrée : position du cible1, position du cible2, configuration initiale du tronc
 - 2: Sortie : configuration du tronc
 - 3: $D1 = |Position\ de\ la\ stereno-clavicule - Position\ du\ poignet|$
 - 4: $D = |Position\ de\ la\ stereno-clavicule - Position\ du\ cible1|$
 - 5: $D2 = |Position\ de\ la\ stereno-clavicule - Position\ du\ poignetd|$
 - 6: $D3 = |Position\ de\ la\ stereno-clavicule - Position\ du\ cible2|$
 - 7: Si($D1 < D$) Calculer la nouvelle position du stereno-clavicul pos1
 - 8: Si($D2 < D3$) Calculer la nouvelle position du stereno-clavicul pos2
 - 9: Calculer la nouvelle position du stereno-clavicule Pos
 - 10: Appliquer PIKOV afin de trouver la nouvelle configuration du tronc qui permet que la stereno clavicule soit dans la position POS sans changer la position de la racine
 - 11: Fin
-

4.8.2.4 Ajustement des membres supérieure et inférieures

Une fois les ajustements du tronc et de la racine sont terminés, nous appliquons la cinématique inverse afin d'ajuster les bras et les jambes selon les positions souhaitées de leur extrémités.

Pour les bras, la racine de la cinématique inverse est le stéréno-clavicule, son extrémité est le poignet, alors que pour les jambes, la racine de la cinématique inverse appliquée est la racine du personnage qui est le pelvis et l'extrémité est le pied.

Algorithme 5 Algorithme d'ajustement du tronc

- 1: Entrée : Position de la cible, Positions des sub-base déjà calculé
 - 2: Sortie : Configuration du bras et des jambes.
 - 3: Appliquer PIKOV sur chaque membre afin de trouver leur nouvelle configuration
 - 4: Fin
-

4.9 Continuité des résultats

La cinématique inverse permet de modifier le mouvement posture par posture, elle ne prends pas en compte les relations entre les frames. Cependant, cela peut produire des discontinuités dans le mouvement. Lecallenc [64] et Cavelho [22] ont utilisés une méthode de filtrage afin de garantir la continuité du mouvement.

Notre technique de modification du mouvement se base sur une méthode rapide de cinématique inverse. Donc, lors de la modification de mouvement nous n'avons pas besoin d'appliquer le filtrage.

Cependant, des situations où l'animateur veut propager une modification faite sur une

posture sur d'autres en spécifiant l'intervalle de la propagation.

Supposons nous que nous avons un mouvement à modifier $MF_1..F_n$, nous voulons modifier la frame F_4 et la frame F_i , figure 4.13. L'intervalle de propagation est $[F_i, F_k]$.

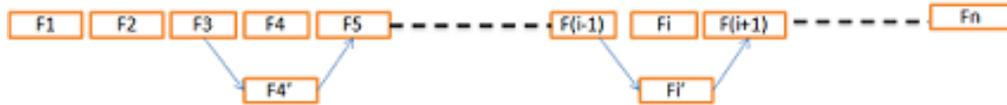


FIGURE 4.13 – Propagation des modifications

- De $F(i-1) \rightarrow F_i$: prendre en compte :
 -
 - Appliquer la cinématique inverse
 - La vitesse de mouvement de référence(garder la même vitesse)
- Dans l'intervalle de modification
 - De $F_i \rightarrow F_k$ doit propager les mêmes configurations sur la partie modifiée dans F_i . Exemple : dans le frame F_i nous avons modifier la partie basse du personnage, dans les frame $F_i+1, F_i+2..F_k$ la configuration de la partie basse de l'humanoïde est la même que dans F_i .
- de $F_k \rightarrow F(k+1)$:
 - Utiliser la cinématique inverse
 - La vitesse de mouvement de référence(garder l

4.10 Résultats

Dans cette section, nous allons présenter quelques résultats de notre méthode d'ajustement. Nos tests ont été effectués sur une machine dotée d'un processeur intel(R)Core(TM)i3-2330M CPU @ 2.20 GHz et sous windows7 comme système d'exploitation.

Les figures 4.14,4.15 représentent respectivement l'atteinte d'un objet avec le bras seulement, et avec le tronc et le bras, en spécifiant la posture initiale du personnage virtuel ainsi que la cible visée(en rouge).

Dans le premier test 4.14, 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.

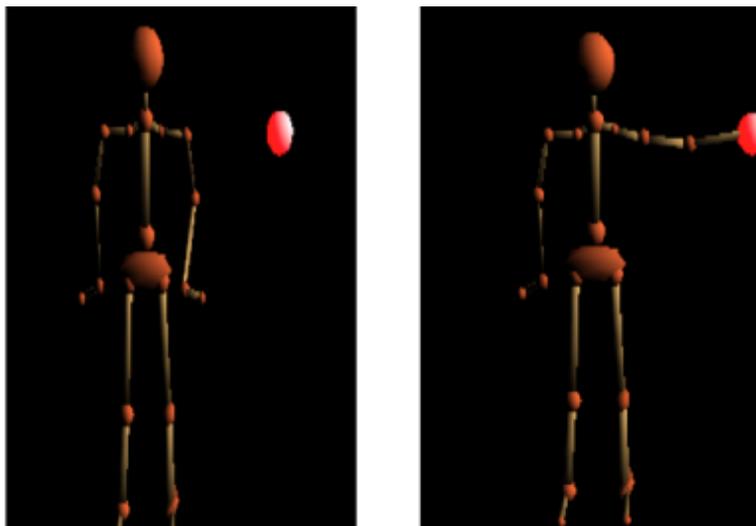


FIGURE 4.14 – Atteindre un objet seulement avec le bras

Nous avons procédé un autre test, où la cible est éloigné du personnage. Dans ce cas, le tronc change de configuration afin d'aider le bras pour atteindre la cible.

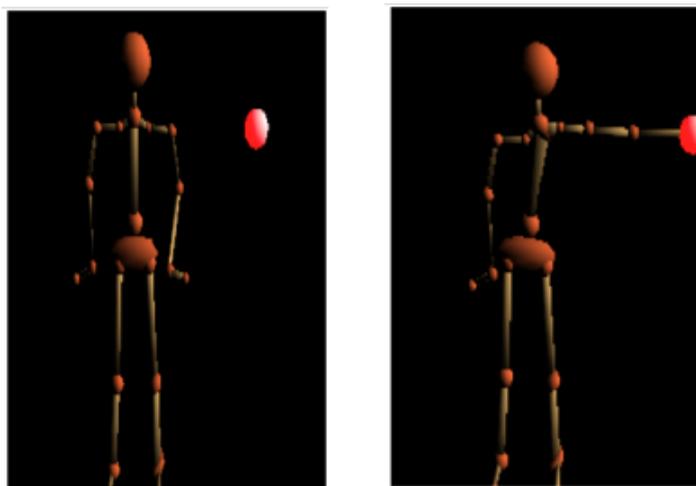


FIGURE 4.15 – Atteindre un objet utilisant le tronc et le bras

4.10.0.5 Multiple contraintes

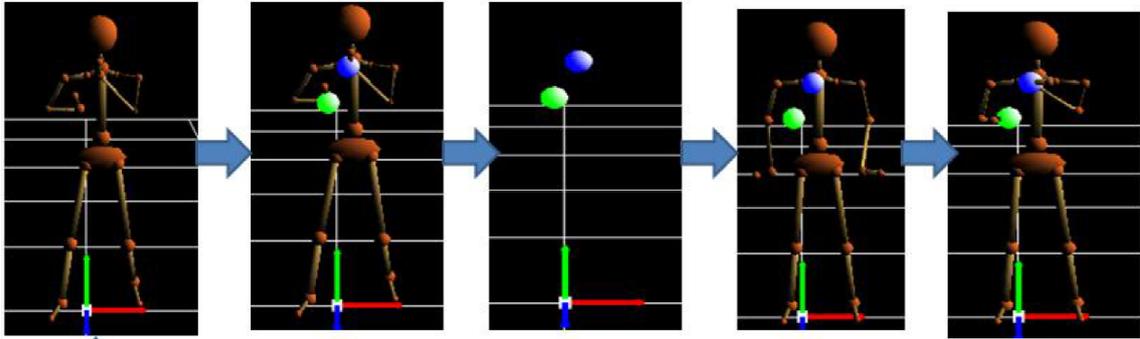


FIGURE 4.16 – Multiple cible à atteindre

Afin de discuter la performance de notre méthode proposée, dans les figures 4.16 et 4.17, nous avons comparé les résultats de notre méthode avec le résultat d'une posture naturelle capturée. Dans le premier test nous avons utilisé la position d'une extrémité d'un bras provenant d'un mouvement naturel déjà capturé. Ceci est considéré comme une position souhaitée de la main de notre personnage virtuel. Nous suivrons ces étapes :

1. Récupération des positions des extrémités d'une posture naturelle déjà capturé.
2. représentation de ces positions par des sphères.
3. Sur un autre plan, nous avons notre personnage virtuel qui est dans une posture initiale,
4. L'application d'une méthode d'ajustement pour que les extrémités du bras du personnage atteignent les sphères.

En comparant les résultats avec la posture naturelle, nous déduisons que c'est presque la même posture.

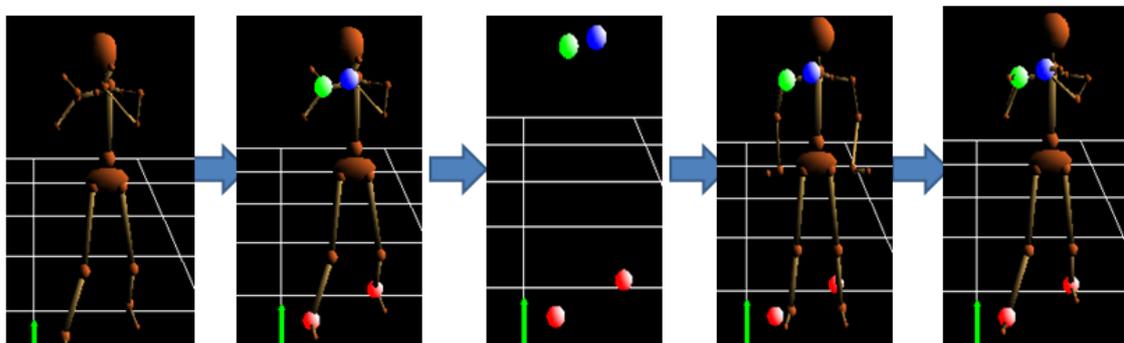


FIGURE 4.17 – Quatre cibles à atteindre

Dans le deuxième test, nous avons quatre tâches à atteindre. Nous récupérons les positions des extrémités des bras et des jambes d'une posture initiale. Ensuite nous les représentons par des sphères. Enfin, nous appliquons la méthode d'ajustement sur notre personnage qui est dans son état initial dont le but que ses quatre extrémités - mains et pieds- soient dans leur positions souhaitées. Nous pouvons remarquer que les résultats de l'ajustement sont proches de la posture naturelle.

Nous avons appliqué la même étape sur toute le mouvement. La figure... montre un échantillon des résultats.

TABLE 4.2 – Comparaison entre une posture capturé et une autre construite par la méthode proposée

Angle	Posture naturelle en rd	Posture construite en rd
Ang1_Arm1	0.5311	0.5551
Ang2_Arm1	0.0003	0
Ang3_Arm1	1.4031	0.3182

Le tableau 4.5 représente une comparaison entre les angles de jointure d'une posture initiale et celle d'une posture finale de notre méthode d'ajustement où ils ont les même positions des extrémités des bras. Depuis la table, nous constatons que les angles d'une posture initiale comparés à ceux de notre posture sont presque les mêmes, cela est due à l'utilisation d'une méthode de modification à base de PIKOV qui prends en charge le respect des limites articulaires.

4.11 Comparaison avec d'autres méthodes

En comparant les résultats avec celui du travail de Kulpa nous pouvons constaté la suite :

- Kulpa dans son travail a utilisé la méthode CCD qui appartient à la classe des méthodes itératives et heuristiques. Mais la méthode FABRIK est plus rapide, dans un autre plan, et d'après les résultats présenté dans le chapitre précédent, notre méthode est plus rapide de la méthode FABRIK. On peut conclure que notre méthode est plus rapide que la méthode de Kulpa.
- La méthode Kulpa ne prend pas en charge la synergie entre les groupes ce qui peut donner des résultats irréalistes, dont notre travail nous évitons ce problème en tenant compte des relations entre les parties du corps

- Le travail de Kulpa manque de coopération entre les groupes lors du traitement de conflits entre les tâches : des solutions peuvent exister mais le partitionnement empêche leur émergence. Dans notre travail est afin de résoudre le conflit nous prenons en compte les deux parties qui sont reliés par un seul jointure. Benoit [65] a proposé aussi une méthode pour la modification interactive du mouvement, sa méthode est différente de la notre par :
 - Elle est basée sur la méthode PIK de cinématique inverse afin de traiter le problème de conflit entre les tâches. Cette dernière se base sur la méthode Jacobien qui est lente comparée avec le principe de fonctionnement de la méthode PIKOV que nous utilisons.
 - C'est une méthode de modification hors ligne, alors que dans notre cas nous pouvons utiliser la méthode en ligne.
 - dans le but d'assurer la continuité entre les postures du mouvement après la modification, il a ajouté une étape de filtrage. Par contre, les résultats de notre méthode sont continue et n'utilise aucun type de filtrage.

4.12 Conclusion

La modification d'un mouvement est une tâche très essentielle dans plusieurs domaines d'application. Dans le cadre de notre projet, nous voulons modifier un mouvement de base afin de créer un autre afin d'aider l'animateur à placer les caméras dans un système de capture de mouvement, un tel système n'existe pas. Nous avons présenté notre proposition qui sert à développer un système de modification du mouvement interactive 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 différentes morphologies, alors que la deuxième sert à modifier interactivement le mouvement. La technique présentée est conçue pour la modification du mouvement que la génération des mouvements.

Dans ce chapitre, nous avons décrit une nouvelle méthode de modification cinématique des mouvements, qui permet de modifier un mouvement d'une manière interactive. La méthode proposée se base sur la méthode PIKOV qui est une méthode de cinématique inverse rapide. La technique présentée dans ce chapitre est conçue pour la modification du mouvement plutôt que de créer un nouveau mouvement.

L'idée principale de la méthode est de diviser le personnage virtuel en groupe relié entre eux par des jointures afin de garantir la corrélation entre les groupes du personnage

4.12. CONCLUSION

ensuite nous appliquons la méthode PIKOV dans un ordre spécifique. Nous avons testé la méthode sur différents cas, la division du personnage en groupe facilite le calcul de la solution. Aussi, choisir des jointures de connexion entre les groupes et appliqué la cinématique inverse dans un ordre définie permet de respecter l'ordre des segments et donne des postures plus réaliste.

Dans le prochain chapitre, nous présenterons notre proposition pour l'adaptation de mouvement à une autre morphologie.

Adaptation des mouvements à une autre morphologie utilisant PIKOV

5.1 Introduction

Nous voulons proposer un outil qui permet d'appliquer différents mouvements sur un personnage virtuel, ensuite de modifier ce mouvement d'une manière interactive. Cependant, les fichiers ne respectent pas tous la même représentation. Ce qui pose beaucoup de difficultés.

Dans ce chapitre nous proposons une méthode d'adaptation d'un mouvement d'un personnage à un autre dont la différence réside dans la topologie et la géométrie en utilisant la cinématique inverse.

5.2 Adaptation d'un mouvement à une autre morphologie

La capture du mouvement nous permet d'obtenir la configuration articulaire de la personne ainsi que la position de la racine de ses chaînes articulaires. Néanmoins, recopier directement ces informations sur le squelette de personnage virtuel, risque d'obtenir des anomalies. Par exemple, en cas de contact avec un autre objet de l'environnement -atteindre un objet- si notre personnage virtuel est plus petit que du sujet réel, le personnage va se retrouver suspendu en l'air, figure 5.1

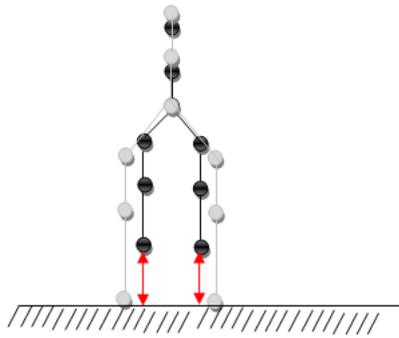


FIGURE 5.1 – Problème de recopie directement d'un mouvement sur un personnage différent

Le principe de l'adaptation de mouvement à une autre morphologie est de transformer des mouvements depuis un squelette canonique à un autre fourni par l'utilisateur. L'adaptation peut être séparée en deux étapes. La première étape, sert à convertir les angles des jointures encodées dans le fichier du mouvement depuis le squelette canonique au squelette de destination. La deuxième étape est de renforcer différentes contraintes des positions tel que la position des pieds et le contact avec des objets dans l'environnement.

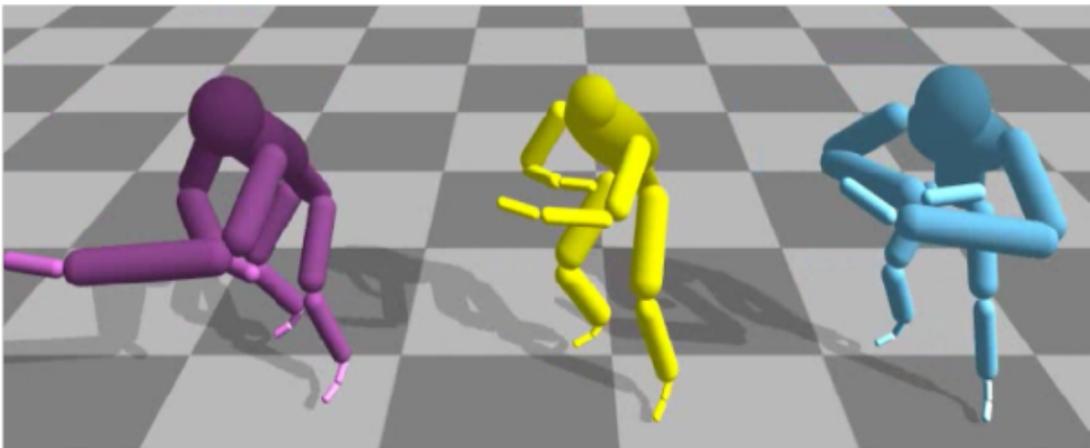


FIGURE 5.2 – Adaptation de mouvement aux d'autres morphologies : le personnage au milieu, représente le personnage du mouvement source. Le personnage à gauche a des membres longs que le vrai acteur le mouvement est transféré en gardant les mêmes angles des jointures. Le mouvement de personnage à droite, est obtenu en utilisant une méthode plus avancée. [42]

Afin d'adapter un mouvement à une autre morphologie, il est préférable d'utiliser des mouvements annotés avec les contraintes à respecter, tel que les périodes de contacts entre

les extrémités et les objets. Cependant, ce n'est pas tous les fichiers du mouvement sont annotés à cause de la difficulté de l'étape d'annotation et la nécessité d'un expert pour la réaliser, l'annotation doit être faite lors de l'acquisition du mouvement.

Des techniques effectives utilisées pour adapter un mouvement à une autre morphologie sont les méthodes spatiotemporelles [35] qui donnent la possibilité d'adapter un mouvement tout en préservant quelques contraintes. Ce sont des méthodes intra-frame qui traitent le mouvement comme un seul bloc. Par contre, elles souffrent du problème de temps de calcul élevé, ce qui limite leur utilisation dans les applications interactives. Afin de surmonter ce problème, les algorithmes de la cinématique inverse sont utilisées aussi afin de préserver les contraintes du mouvement d'origine et d'appliquer d'autres contraintes en temps réel.

Choi et Ko [24] ont présenté le premier travail pour l'adaptation de mouvement à une autre morphologie en temps réel. Ils ont construit un solveur de la cinématique inverse en se basant sur une méthode de contrôle de la vitesse du mouvement. La tâche principale sert à suivre le mouvement des extrémités du personnage. La deuxième tâche, est de recopier le mouvement du personnage de source autant que possible. Les résultats obtenus permet de renforcer la cohérence entre les frames implicitement. Kulpa [62] figure 5.3 a proposé une notation indépendante de la taille du squelette ce qui lui permet d'appliquer facilement les adaptations. Son système préserve les positions des extrémités au lieu des angles. Il a utilisé une extension de la méthode de la cinématique inverse CCD afin d'appliquer les adaptations sur les parties du corps de personnage simultanément.

Cependant, toutes ces méthodes supposent que la différence entre les deux squelettes -source et destination- soit seulement par rapport à la taille des segments. Hecker [41] a proposé un outil qui permet d'adapter un mouvement entre des personnages non humain se basant sur une méthode de la cinématique inverse à base de particules. Cette méthode souffre des collisions et de discontinuité lorsqu'elle est appliquée à des scènes avec des interactions entre les personnages.



FIGURE 5.3 – Adaptation du mouvement utilisant un squelette indépendant.[62]

Monzani et al [75] ont utilisé un squelette intermédiaire afin d'adapter un mouvement à un autre personnage, les différences entre les deux personnages résident dans la géométrie et la topologie. La géométrie spécifie la taille des segments tandis que la topologie spécifie le nombre des jointures. La première étape du travail de Monzani sert à spécifier les correspondances entre les articulations du personnage source (Performer Skeleton) et celle du squelette destination (end user skeleton). La figure 5.4 à gauche montre cette étape pour les articulations d'un bras. Dans la deuxième étape, les positions sont mises en correspondance afin de transposer les orientations du mouvement initial sur le nouveau personnage (figure 5.4 à droite.). Dans la dernière étape, Monzani a spécifié des contraintes cinématiques afin de conserver certaines caractéristiques du mouvement d'origine, la résolution des contraintes est faite par l'utilisation de la cinématique inverse. Afin d'assurer la continuité du mouvement tout en respectant les contraintes, il a ajouté des périodes « ease-in (eas out) avant (respectivement après) que les contraintes soient activées (respectivement désactivées). Durant les périodes ease in les trajectoires des extrémités sont interpolées linéairement afin d'atteindre les objectifs spécifiés d'une manière continue. Durant les périodes ease out, il a interpolé linéairement les postures au lieu d'interpoler les trajectoires. Cette méthode donne de bons résultats dans le cas où le mouvement adapté est proche du mouvement original, l'interpolation linéaire peut produire des discontinuités parce qu'elle ne prend pas en compte la vitesse des extrémités. Cette méthode a aussi besoin d'un espace de stockage additionnel afin de sauvegarder les segments du milieu.

Komura [57] a proposé d'adapter un mouvement capturé à une nouvelle physionomie. Cette méthode est capable d'adapter la taille du personnage ainsi que l'énergie pouvant être mise en œuvre par une personne. Dans [58], les auteurs ont appliqué cette méthode

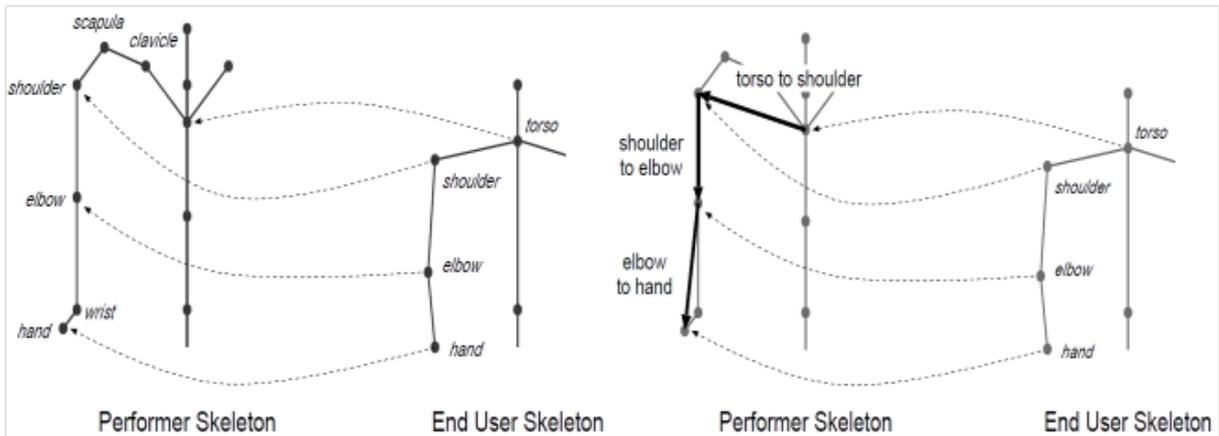


FIGURE 5.4 – Adaptation de mouvement utilisant un squelette intermédiaire.[75]

pour l'adaptation du mouvement de la marche.



FIGURE 5.5 – Adaptation de mouvement à une nouvelle physionomie[57].

5.3 Bilan et proposition

Une méthode d'adaptation des mouvements à différentes morphologies est nécessaire dans un outil de modification des mouvement afin de permettre à l'utilisateur d'importer et de modifier plusieurs mouvements. Une méthode efficace d'adaptation des mouvements doit être :

- Utilisée en temps réel où on applique la méthode lors de la lecture de fichier du mouvement.
- Appliquée sur des personnages avec différentes topologies et différentes tailles des segments.

Les méthodes d'adaptation basées sur la cinématique inverse sont les plus utilisées ceci est dû à la possibilité de définir des contraintes à préserver dans le mouvement source. Néanmoins, la qualité et la rapidité des résultats dépendent de la méthode de la cinématique inverse utilisée.

Le travail de Kulpa donne des résultats rapides mais sa méthode de IK ne prend pas en compte les relations entre les parties du corps lors de l'adaptation de plusieurs segments simultanément.

Dans un autre plan, malgré la qualité des résultats utilisant la méthode de Monzani, l'utilisation d'un squelette intermédiaire avec la cinématique inverse augmente le temps de calcul et l'espace de stockage, tandis que la négligence des jointures qui n'ont pas de correspondance risque d'affecter la qualité des résultats.

Dans ce travail, nous avons proposé un outil qui permet d'adapter un mouvement à différentes topologies et géométries en utilisant une méthode rapide de la cinématique inverse PIKOV appliquée à chaque frame. La méthode permet de surmonter le problème posé dans le travail de Kulpa par la spécification des relations entre les parties du corps du personnage afin de garantir la synergie entre les différentes parties et augmenter le réalisme des résultats. En outre, la méthode de la cinématique inverse utilisée est rapide est cela résout le problème de discontinuité. Elle permet aussi de surmonter le problème de la méthode de Monzani en proposant un algorithme qui permet de calculer les configurations des jointures n'ayant pas de correspondances dans le fichier du mouvement.

5.4 Aperçu du système

La figure 5.6 représente une vue générale du système proposé pour l'adaptation du mouvement à d'autres morphologies. Le personnage virtuel est modélisé indépendamment du fichier de mouvement, cela donne l'avantage d'appliquer plusieurs fichiers de mouvement sur un seul personnage. Il se base sur une méthode de la cinématique inverse rapide, qui traite le problème de conflit entre les contraintes et le problème d'évitement d'obstacles lors de l'adaptation.

La première étape sert à importer un fichier de mouvement depuis la base. Une fois l'importation faite, nous allons appliquer une méthode de mise en correspondance entre les informations des jointures depuis le fichier de mouvement et notre personnage virtuel. Afin de diminuer l'intervention des utilisateurs et mettre les pieds du personnage sur le sol, nous calculons automatiquement la position des pieds en prenant en compte la différence

5.5. ADAPTATION DES MOUVEMENTS UTILISANT LA CINÉMATIQUE INVERSE

entre les deux squelettes. En outre, nous utilisons une méthode de la cinématique inverse afin de préserver certaines caractéristiques essentielles du mouvement d'origine tel que le contact avec d'autres objets.

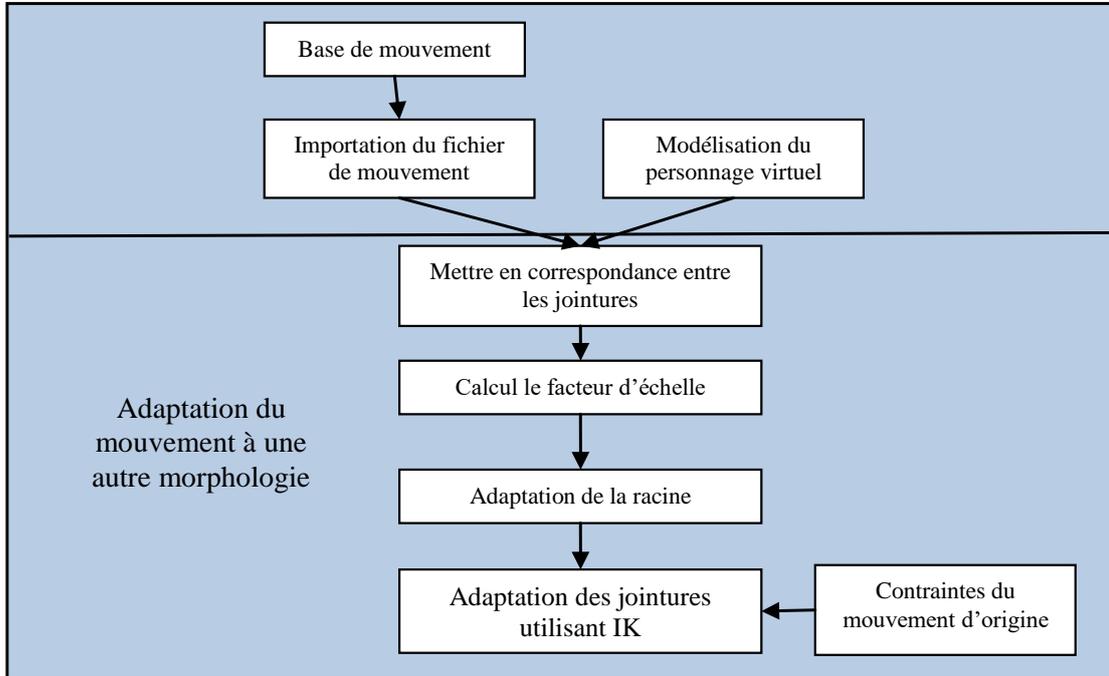


FIGURE 5.6 – Schéma général de système d'adaptation des mouvements

5.5 Adaptation des mouvements utilisant la cinématique inverse

Afin de résoudre le problème d'adaptation, nous allons introduire une nouvelle méthode per-frame en se basant sur les positions des extrémités. Nous voulons dire par "per-frame" l'application de la méthode à chaque frame du mouvement d'origine.

L'objectif est d'adapter un mouvement déjà capturé d'un personnage à un autre, ayant différentes topologies et différentes géométries. Pour mieux représenter la méthode proposée, nous allons commencer par la présentation du fichier de mouvement puis nous allons détailler les modules de systèmes.

5.5.1 Structure du fichier de mouvement

Comme étape initiale, nous avons utilisé des fichiers du mouvement de format BVH. Les informations représentent les détails des relations de type père/fils des segments et

les angles d'Euler de chaque jointure dans chaque frame.

Le mouvement, représente une variation de la morphologie au cours du temps. La morphologie du personnage virtuel est exprimée dans la posture de son squelette, qui est une hiérarchie de jointures connectées avec des neuds. Chaque jointure a deux informations, l'orientation et l'offset qui est la distance entre ce dernier et sa jointure père. Pour la racine, l'offset représente la position du personnage dans l'espace. Le squelette est défini par :

$S=(j_0,..j_n)$ avec $j_0=(p,r)$ et $j_i=(x,p,r)$ pour tout autre i

Où

j représente le jointure et i son indice,

x est l'indice de jointure père.

P représente l'offset et r est la rotation.

Le mouvement d'un squelette est défini comme une fonction qui donne la position de la racine et les rotations des jointures à chaque pas de temps.

$$M(t) = (p(0, t), r(0, t), \dots, r(n, t)) \quad (5.1)$$

Où $M(t)$ c'est la posture du personnage à l'instant t , $p(0,t)$ la position de la racine et $r(i,t)$ est la rotation de jointure j_i dans l'instant t .

Les rotations dans le fichier de mouvement sont représentées par des angles d'Euler. Néanmoins, vue l'efficacité de l'utilisation des quaternions pour le contrôle des personnages virtuels, nous avons proposé de convertir les angles de fichier de mouvement en quaternion. Afin de calculer le quaternion, dans un premier pas, nous calculons une matrice 3*3, ensuite nous convertissons la matrice en quaternion.

Afin d'adapter le mouvement, nous avons choisi d'utiliser une méthode d'adaptation du mouvement à base de la cinématique inverse, qui permet la mise à jour de la posture dans chaque frame ce qui implique de respecter les contraintes du mouvement d'origine.

5.5.2 Mise en correspondance

Cette étape est essentielle dans tous les systèmes d'adaptation du mouvement. Des travaux tel que [31] ont utilisé une méthode heuristique afin de copier les informations entre le personnage source et destination. Monzani [75] a utilisé un squelette intermédiaire afin de réaliser cette étape. Cependant, la plupart des systèmes d'adaptation de mouvement le font manuellement.

5.5. ADAPTATION DES MOUVEMENTS UTILISANT LA CINÉMATIQUE INVERSE

Dans notre proposition pour cette étape, nous avons choisi d'appliquer un algorithme qui se base sur les noms des jointures.

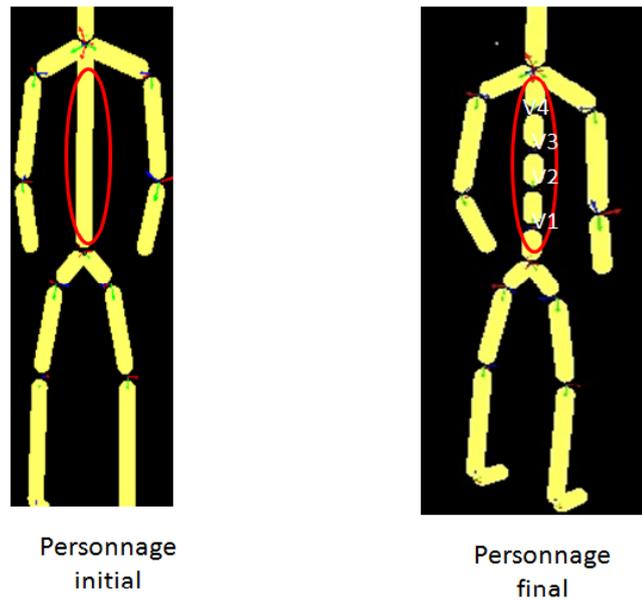


FIGURE 5.7 – Deux squelettes avec différents tronc

Notant que les deux squelettes n'ont pas le même nombre des jointures tel que présenté dans la figure 5.7 ni le même indice des jointures, la seule hypothèse que nous avons émise est que les jointures que l'on retrouve dans les deux squelettes ont les mêmes noms, c'est pourquoi nous renommons ces jointures dans le fichier de mouvement à l'aide d'une interface figure 5.8.



FIGURE 5.8 – Renommer les jointures communes entre le fichier de mouvement et le personnage virtuel

5.5. ADAPTATION DES MOUVEMENTS UTILISANT LA CINÉMATIQUE INVERSE

Afin de recopier les angles d'orientation depuis le fichier de mouvement à notre personnage virtuel nous avons proposé un l'algorithme 6 qui permet de chercher des jointures de squelette d'origine selon les noms et de copier leur angles dans le squelette final.

Algorithme 6 Algorithme de mise en correspondance entre les jointures de fichier du mouvement et le personnage virtuel

```
1: nombre=nombre de jointure de fichier de mouvement
2: Pour toutes les jointures de notre squelette de i=0..n
3: Num <-0
4: trouver<- faux ;
5: Tantque ((num < nombre ) et (trouver=faux ))
    //bonename, nom de jointure de squellete
6: si ((bonename!= nom[num] ))
7: num=num+1 ;
8: sinon trouver<- vrais ;
9: FinSi
10: SI(trouver=vrais)
11: int in=(3*(num))+3;// pour éviter les informations de la racine
12: orientation[num] <-orientation[in] depuis le fichier de mouvement
13: FinSi
14: FinTantque
15: Fin
```

Afin de trouver les configurations des jointures de notre personnage virtuel sans correspondance, nous utiliserons la méthode PIKOV de la cinématique inverse.

5.5.3 Adaptation du tronc et du cou

La figure 5.9 montre un exemple de l'étape de mise en correspondance entre le tronc du mouvement d'origine (gauche) et du possible tronc du personnage destination(droite). Après l'étape de la mise en correspondance entre les noms des jointures, nous trouverons qu'il existe des jointures du tronc du nouveau personnage sans correspondances. Pour cela, nous avons proposé une solution à ce problème en alignant les débuts des deux troncs, ensuite, les extrémités du tronc doivent également être alignées en calculant les rotations de toutes les jointures intermédiaires du tronc. Afin de calculer les rotations de toutes les jointures du tronc, une méthode de la cinématique inverse est appliquée.

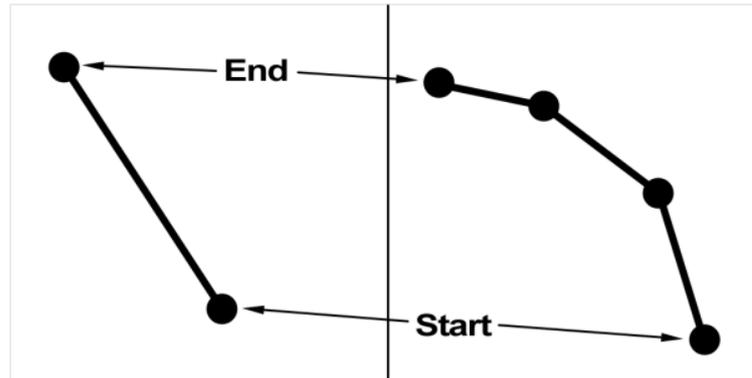


FIGURE 5.9 – Adaptation du tronc

5.5.4 Adaptation de la racine

Les postures du squelette source doivent correspondre exactement aux postures du mouvement que nous allons utiliser pour notre simulation. Même si aucune contrainte n'est appliquée, il faut adapter les postures au sol.

Étant donné que le positionnement du personnage virtuel dépend directement de la racine du squelette, la position et l'orientation de la racine du squelette restent deux données importantes. Néanmoins, pour l'animation des personnages virtuels, appliquer la même position de la racine sur différents squelettes peut rompre les contacts des pieds avec le sol. Par exemple, en appliquant le même mouvement d'un adulte à un enfant, les pieds de ce dernier seront suspendus en air, figure 5.10.a. Dans la figure 5.10.b, nous avons appliqué la même position de la racine d'un petit personnage sur un autre plus grand, les pieds sont sous sol.

5.5. ADAPTATION DES MOUVEMENTS UTILISANT LA CINÉMATIQUE INVERSE

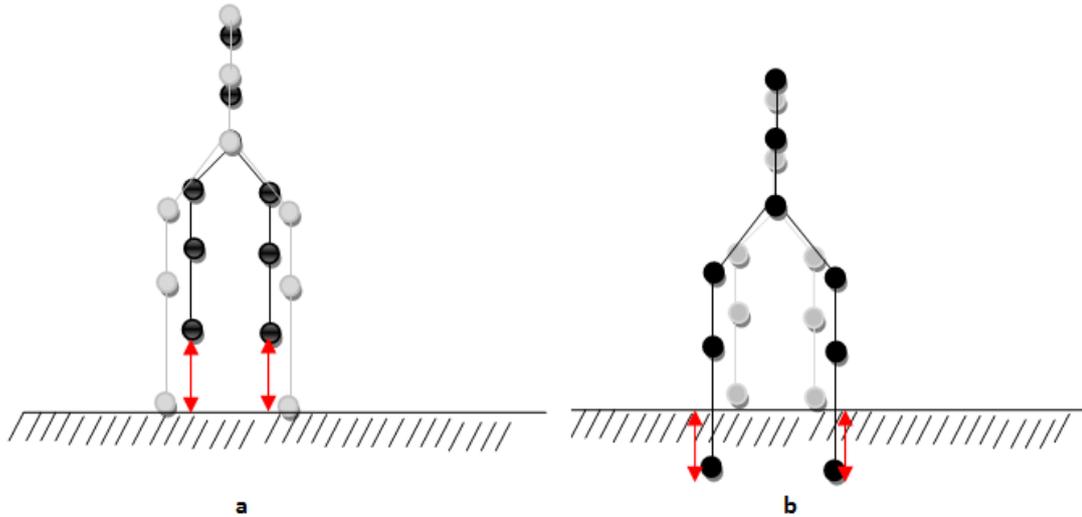


FIGURE 5.10 – Problème lors de l’application de la même position de la racine sur différents personnage qui n’ont pas la même taille : dans la figure en gauche, le personnage de destination est petit. La figure en droite montre le résultat si on applique la position d’un personnage de petit taille à un autre plus grands.

Une étape d’adaptation de la racine est nécessaire lors de l’application du mouvement à une autre morphologie. Cette adaptation répond à deux critères :

- La nouvelle position de la racine doit tenir compte de la morphologie du personnage et en particulier les longueurs des jambes.
- Les jambes doivent être adaptées pour respecter cette nouvelle position de la racine.

Pour cela, une première étape consiste à calculer le facteur d’échelle, qui est la portion entre la longueur des jambes du personnage final et les longueurs des jambes du squelette d’origine.

$$\alpha = l_1/l_0\dots \quad (5.2)$$

Ensuite nous calculons la nouvelle position de la racine en nous basant sur deux éléments : le facteur d’échelle et la position de la racine dans le fichier de mouvement. La nouvelle position de racine est calculée par l’équation

$$p_1 = (p_0 - d) * \alpha + d.. \quad (5.3)$$

Où p_o est la position initiale de la racine dans le fichier du mouvement, d est la valeur de l’hauteur de sol et si le personnage marche sur un objet c’est l’hauteur de cette objet, et α c’est le facteur d’échelle.

5.5.5 Adaptation des jointures

Après la mise en correspondance entre les angles des jointures depuis le fichier de mouvement et les jointures de notre personnage virtuel, à chaque frame nous vérifions si les extrémités ont besoin de répondre à certaines contraintes ou non.

$$f(M(t_j)) = P_j, t_j \in T_j \quad (5.4)$$

$T_j=[t_{jd},t_{jf}]$ est l'intervalle de temps de contrainte et p_j est la position de l'extrémité à contraindre.

Dans le cas positif, nous appliquons la méthode de la cinématique inverse PIKOV afin de renforcer l'ensemble des contraintes de position.

$$Mt_j = IK(M(t_j, P_j)) \quad (5.5)$$

Lors de l'existence de plusieurs contraintes en même temps, les positions des pieds sur le sol est la contrainte avec la plus haute priorité et doit être respectée impérativement.

Grâce à la rapidité de la méthode de cinématique inverse utilisée, la continuité du mouvement final est assuré et on est pas besoin d'une autre étape.

5.6 Résultats

Les algorithmes présentés dans ce chapitre sont implémentés et testés sur plusieurs types des mouvements avec différentes contraintes.

5.6.1 Adaptation de la racine

Premièrement, nous importons un fichier de mouvement, ensuite nous appliquons ce mouvement à un personnage qui a les mêmes longueurs des segments que celui du fichier. Par ailleurs, nous appliquons la même donnée sur un autre personnage qui est plus grand que le personnage du fichier de mouvement. Nous obtenons des résultats où les pieds du deuxième personnage ne sont pas sur le sol figure 5.11.a.

Après l'application de l'étape de l'adaptation de la racine, les deux personnages ont les pieds sur le sol, (figure 5.11).

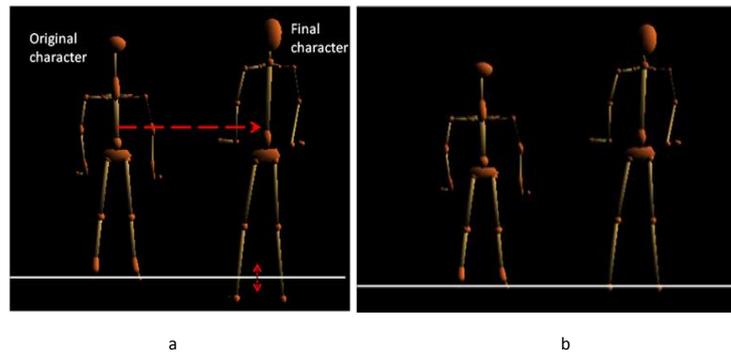


FIGURE 5.11 – Test1 : Adaptation de la racine pour un personnage plus grands que le personnage source

Dans un deuxième test, nous utiliserons un personnage plus petit, sans appliquer l'étape d'adaptation de la racine, le deuxième personnage est suspendue en l'air.

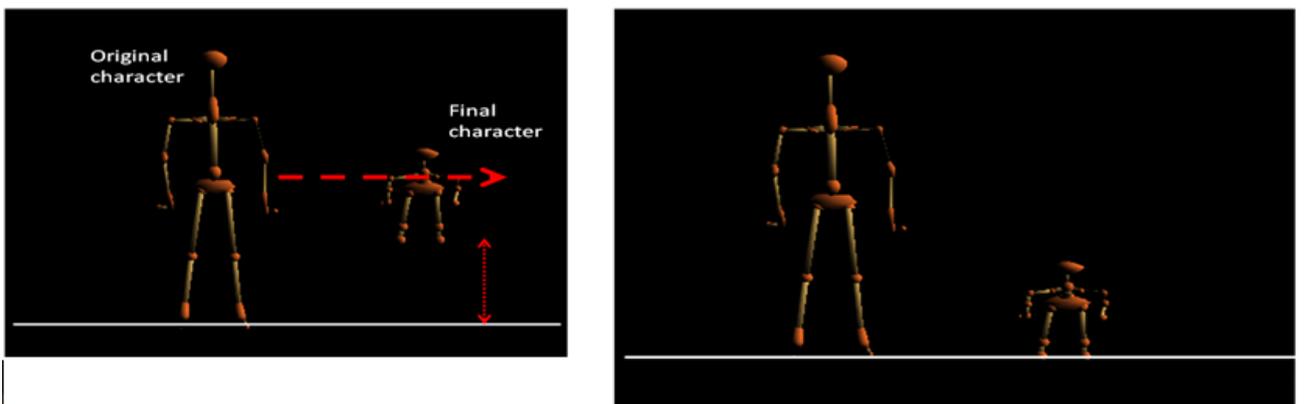


FIGURE 5.12 – Adaptation de la racine pour un personnage plus petit que le personnage source. La figure à gauche représente l'application de la même position de la racine sur un personnage plus petit. La figure à droite montre le résultat après l'étape de l'adaptation.

5.6.1.1 Adaptation du mouvement de la marche

Dans cette expérimentation, le mouvement d'origine est un mouvement capturé avec 798 frames. nous avons appliqué le mouvement de la marche sur trois différents personnages virtuels figure 5.13.

Comme la partie inférieur du corps est plus importante dans le mouvement de la marche et comme il n'existe aucune contrainte à préserver, nous allons appliquer la méthode d'adaptation seulement sur la partie inférieure qui consiste de pelvis, cuisse, jambe et pied.

5.6. RÉSULTATS

le personnage de final est réduit, la comparaison entre les dimension est donné dans le tableau 5.1

TABLE 5.1 – Comparaison entre les dimensions du personnage virtuel

Jointure	Origine	Final
pelvis	6	6
cuisse	46	33
jambe	50	29
pied	32	10

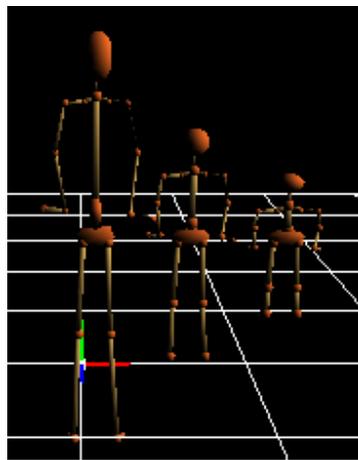
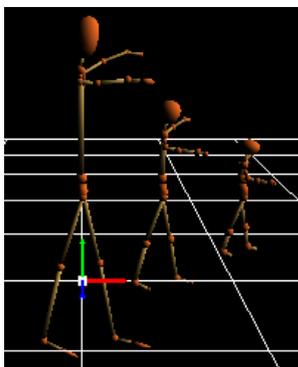
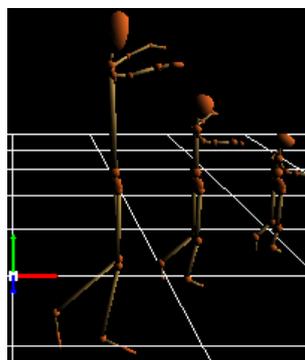


FIGURE 5.13 – Trois personnes utilisés pour adapter le mouvement de la marche

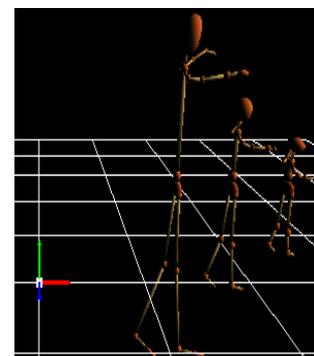
Les figures 5.14 et 5.15 montrent les résultats de l'application de mouvement de la marche sur différents personnages ayant différents tailles des segments.



Frame 0



Frame 100



Frame 200

FIGURE 5.14 – Application du même mouvement de la marche sur les trois personnes

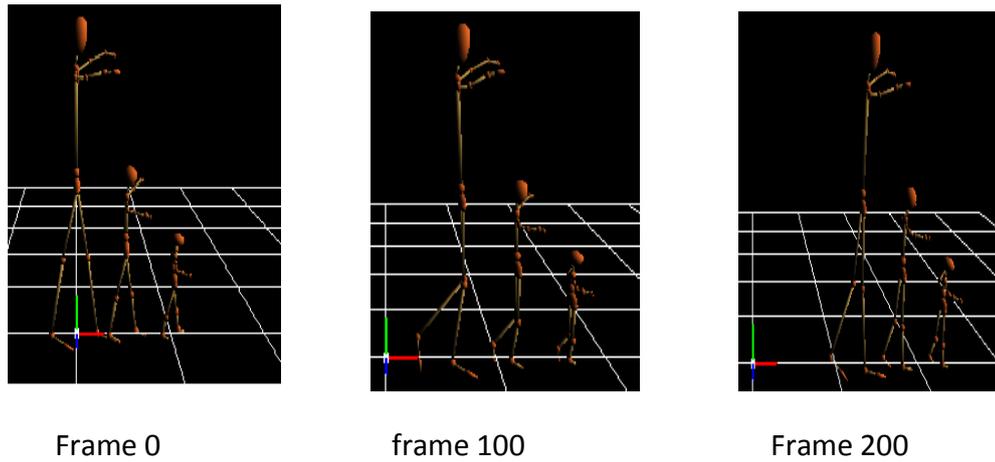


FIGURE 5.15 – Adaptation de la racine des personnage pour un mouvement de la marche

5.6.2 Respect des contraintes

Cette expérimentation a comme but de tester la fiabilité des algorithmes d'adaptation de mouvement à un autre morphologie lors de l'existence des contraintes à respecter.

Nous avons récupéré un mouvement de préhension contient 3200 frames avec un interval de 0.01 de temps entre les frames successives, la figure 5.16 illustre l'application du mouvement à un personnage qui a les mêmes caractéristiques -longueur des segments- que le personnage du mouvement, les pieds sont sur le sol, dans la fin de mouvement(frame 480) le personnage atteint un objet.

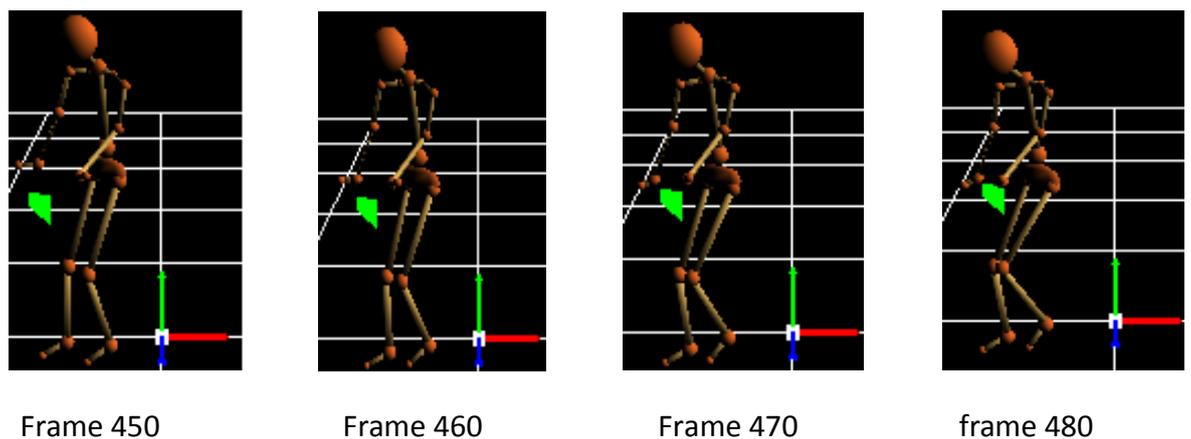


FIGURE 5.16 – Mouvement de préhension d'un objet : le personnage a les mêmes caractéristiques que le personnage d'origine. Les pieds de personnage sont sur le sol et atteint l'objet

Dans la figure 5.17 nous avons appliqué le même mouvement sur un autre personnage qui est plus grand que le personnage d'origine. Nous pouvons remarquer que les pieds de personnage sont en dessous du sol, cela dû à la longueur des jambes qui sont plus long que le personnage d'origine. La figure montre aussi, que le personnage ne peut pas atteindre l'objet atteint par le personnage d'origine.

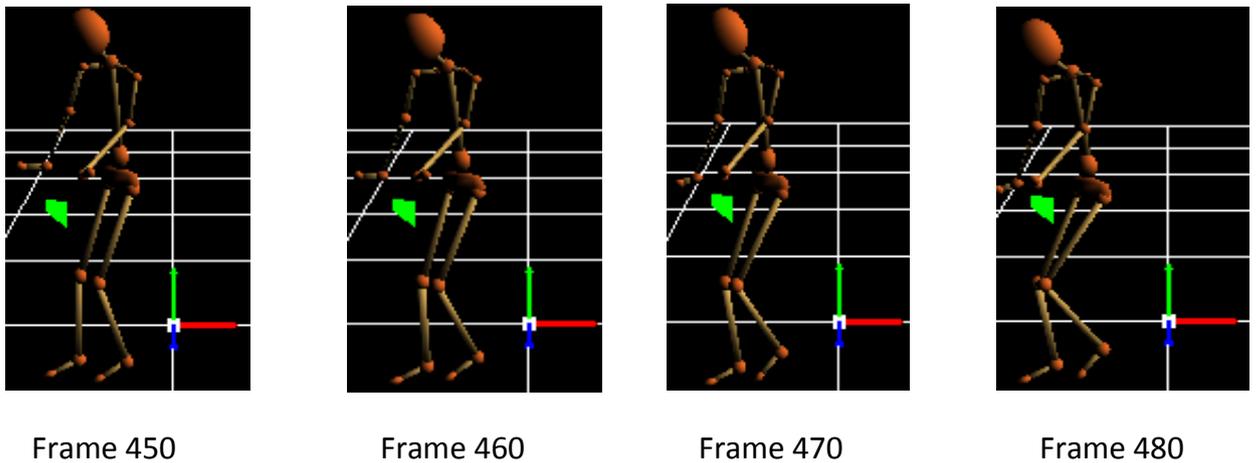


FIGURE 5.17 – Mouvement de préhension d'un objet : le personnage est plus grand que le personnage d'origine. Les pieds de personnage en dessous de sol et le personnage n'atteint pas l'objet

En appliquant seulement l'étape d'adaptation de la racine au sol, les pieds du personnage sont au sol dans la figure 5.18, mais toujours il ne peut pas atteindre l'objet. Afin de calculer la nouvelle position de la racine, nous avons calculer le facteur d'échelle qui est 2, dans ce test l'hauteur de sol est égal à zéro.

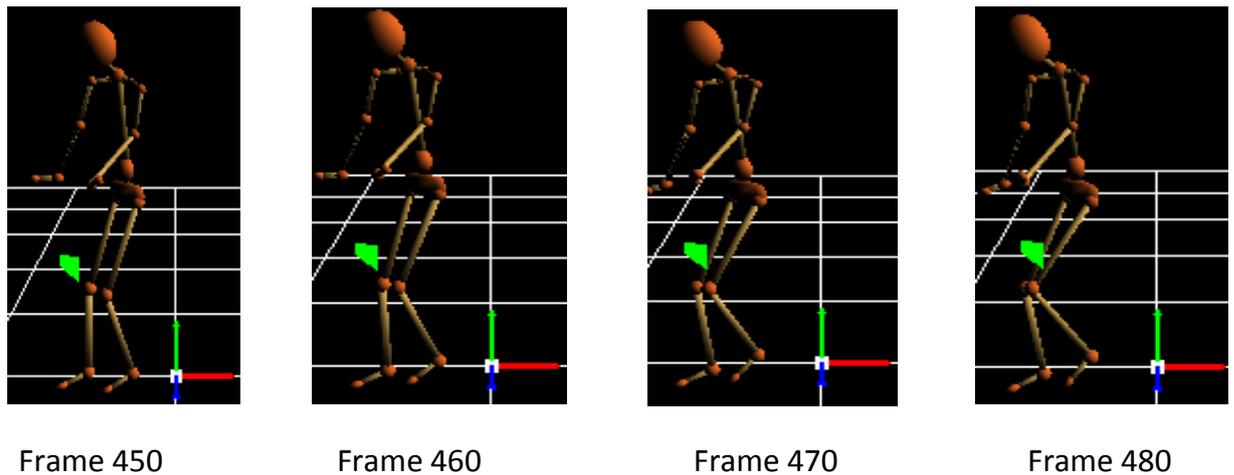


FIGURE 5.18 – Application seulement l'étape d'adaptation de la racine : Les pieds sont au sol mais le personnage ne peut pas atteindre l'objet

Nous avons préservé les contraintes de positions des extrémités -mains- dans le frame 480 de personnage d'origine afin de les utiliser comme des contraintes de position sur les extrémités du personnage final. La figure 5.19 montre les résultats après l'application de la méthode cinématique inverse dans la frame 480 afin de préserver les contraintes déjà définie, le personnage peut atteindre l'objet dont le tronc a contribué dans le mouvement.

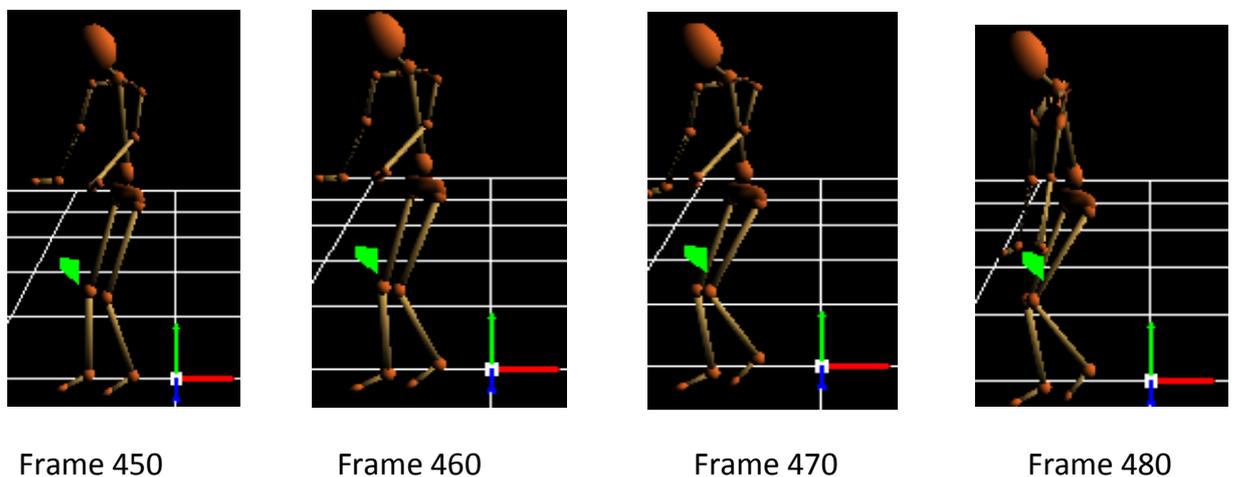


FIGURE 5.19 – application de la méthode PIKOV afin de préserver les contraintes de position, le personnage atteindre l'objet

Une comparaison entre la trajectoire de la main d'un personnage d'origine et un personnage plus grand est donné dans la figure 5.20, nous avons activé des contraintes sur la main du personnage de destination depuis la frame 480. Après l'activation des contraintes,

le trajectoire pointillée qui représente la trajectoire de la main de personnage de destination est exactement la même trajectoire de la main de personnage d'origine.

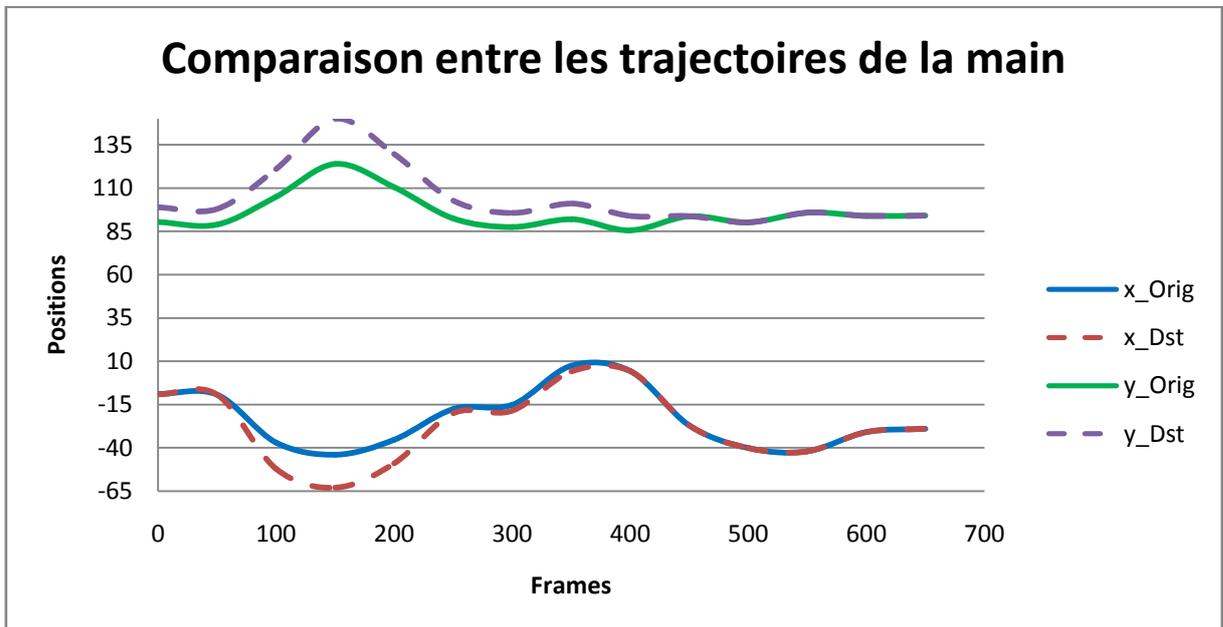


FIGURE 5.20 – Graphe de comparaison entre la trajectoire en x,y de la main de personnage d'origine et celle de personnage de destination

5.7 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode afin de résoudre le problème d'adaptation du mouvement à une autre morphologie en temps réel en utilisant une méthode de la cinématique inverse sans l'utilisation d'une étape supplémentaire tel que le filtrage. Les personnages ont différentes géométries et différentes topologies.

Nous avons proposé une étape d'adaptation de la racine du squelette afin de garantir les positions des membres inférieurs par rapport au sol. Dans le cas où le squelette final a différentes topologies, nous n'ignorons pas les autres articulations inexistantes dans le fichier du mouvement, ni utilisons un autre squelette intermédiaire, mais nous proposons de calculer les rotations de ces articulations en utilisant une méthode de la cinématique inverse.

Dans le cas où il existe des contraintes à préserver dans le mouvement d'origine, nous utilisons notre méthode de cinématique inverse déjà proposée afin de préserver ces contraintes.

Conclusion générale

Les systèmes VICON sont des systèmes utilisés pour la capture de mouvement et sont largement utilisés dans la médecine, le cinéma, la simulation, ...etc. Ils permettent à travers un ensemble de caméras et des réflecteurs positionnés sur un objet de capturer son mouvement pour l'étudier ou le modifier. Dans de tels systèmes le positionnement des caméras pose un problème sérieux à cause des occultations des réflecteurs par l'objet lui-même ou par les obstacles pouvant exister dans l'environnement.

Le projet cadre de cette thèse vise à proposer un système permettant de calculer automatiquement le nombre et les positions optimales des caméras utilisées dans une scène VICON. Le travail réalisé dans cette thèse propose des solutions pour modéliser une scène dotée d'un système VICON à savoir les caméras, les réflecteurs, le sujet en mouvement, les obstacles et créer le mouvement souhaité du sujet.

En effet, plusieurs points ont été développés :

1. Création de la scène : un outil de création de scènes virtuelles a été proposé. Il permet de :
 - (a) Ajouter, modifier, repositionner, réorienter ou supprimer des objets et des caméras dans l'environnement.
 - (b) Importer un fichier de mouvement de type BVH ou C3d et lui ajouter les informations de la scène. Cela nous a permis de créer une base de mouvement contenant différents types de mouvements.
 - (c) Modifier un mouvement d'un personnage : l'outil permet d'appliquer le mouvement lu depuis le fichier sur un personnage virtuel. Ce dernier, peut être le même personnage du fichier de mouvement ou un autre n'ayant pas les mêmes caractéristiques.

2. Modification du mouvement : C'est l'étape la plus cruciale dans le travail du projet. 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 répondent aux spécificités de notre cas d'étude. Cependant, ces méthodes utilisent une base de mouvement afin de modifier un mouvement donné, ce qui limite leur rendement étant donné que les mouvements enregistrés dans la base sont toujours liés aux caractéristiques et au type du personnage considéré et leur adaptation à un autre personnage est généralement très difficile. Aussi, les personnes dont on souhaite capturer le mouvement sont généralement de mobilité réduite et de conditions particulières, ce qui complique la possibilité d'obtention des mouvements similaires dans les bases enregistrées.

Pour palier ces lacunes, nous avons proposé une méthode permettant de contrôler des postures de personnages de manière rapide et réaliste. Elle permet d'atteindre plusieurs objectifs simultanément en un temps réduit et sans conflit en introduisant le principe de priorité. Notre méthode prend en charge l'existence des obstacles dans la scène. Les résultats obtenus sur des chaînes cinématiques simples sont encourageants.

(b) Modification interactive : Plusieurs méthodes de contrôle sont proposées dans la littérature. Notre contribution est de s'éloigner des méthodes de contrôle récentes qui sont basées sur l'utilisation d'une base de mouvement. En outre, l'utilisation d'une large base de mouvements pour ces méthodes implique un grand espace de stockage. Elles ne donnent pas aussi les résultats souhaités si les contraintes à vérifier n'existent pas dans la base. Notre contribution, consiste à proposer une méthode de modification de mouvement en :

- i. subdivisant le corps du personnage virtuel en groupes, puis appliquer la méthode de cinématique inverse PIKOV sur chaque groupe.
- ii. proposant des corrélations entre les parties du corps afin d'obtenir des résultats plus réalistes .

(c) Adaptation du mouvement à une autre morphologie : Parmi les avantages de

notre outil de modification de mouvement la possibilité de son utilisation sur différents mouvements, où nous importons des fichiers de mouvements ensuite les modifions d'une manière interactive. Malheureusement, cette importation a un inconvénient majeur : les personnages source et cible n'ont pas les mêmes caractéristiques de morphologie et de longueur des segments. L'application directe du même mouvement sans correction peut produire des anomalies.

Pour palier cette limite, nous avons proposé une méthode d'adaptation du mouvement basée sur une méthode de cinématique inverse rapide. Généralement, les méthodes d'adaptation ne peuvent pas être appliquée enligne parce qu'elles ont besoin d'une autre étape d'interpolation ou de filtrage afin de garantir la continuité du mouvement final. Notre méthode fonctionne enligne parce qu'elle n'a besoin d'aucune étape de filtrage grâce à la rapidité de la cinématique inverse.

Perspectives

Les méthodes présentées dans les chapitres 5 et 4 permettent d'adapter le mouvement à d'autres situations ou d'autres morphologies. Ces méthodes permettent de respecter un certain nombre de contraintes géométriques en utilisant la cinématique. Cependant, cette dernière ne satisfait que les contraintes géométriques sans tenir compte des contraintes physiques. Donc, la posture finale n'est pas toujours réaliste par rapport aux contraintes fournies par l'utilisateur. Nous proposons une méthode qui intègre la cinétique inverse (prendre en considération la position de centre du masse) avec la cinématique inverse. Le concept clé est l'intégration (et non pas l'application) de la méthode de cinématique inverse, parce que la prise en considération de l'aspect physique peut nuire aux contraintes vérifiées par la cinématique inverse.

Nous envisageons aussi d'analyser des mouvements humains afin d'extraire d'autres synergies entre les articulations pour les modéliser dans notre système afin d'obtenir des mouvements plus réaliste et plus naturels.

Nous comptons aussi d'améliorer le type de l'interactivité, où l'utilisateur pourra modifier le mouvement non seulement par la souris mais en utilisant aussi la main par l'intermédiaire de l'écran tactile, ainsi, il aura la possibilité d'utiliser tous ses doigts pour modifier la posture.

Annexe : Architecture logicielle du système proposé

1 Introduction

La problématique générale de notre projet s'appuie sur le positionnement optimal des caméras. Notre sujet est le premier module de ce grand projet, son objectif consiste à développer un outil sert à créer et à modifier interactivement des mouvements. Plusieurs travaux pour créer un système interactif de modifications des mouvements sont proposés, mais selon nos besoins, ils ne sont pas appropriés à cause de :

- L'utilisation des données provenant de plusieurs outils : ils créent le personnage et son animation par différents outils, puis ils les importent à leur système pour effectuer la modification.
- Proposition de solutions pour des problèmes particuliers : marche, saut, chaque travail pour un mouvement particulier, ce sont des systèmes pour des mouvements limités.

Afin de surmonter ces particularités nous proposons :

- D'intégrer tous les processus dans le même système : modélisation du personnage virtuel, l'importation des fichiers de mouvements et l'application de ces mouvements sur le personnage déjà modélisé, ce qui rend notre système indépendant des autres outils commerciaux.
- de généraliser le système d'édition de mouvements : Dans notre système, nous ne limitons pas le type de mouvement. Il est destiné à modifier n'importe quel mouvement.
- Proposer un outil qui mette à la disposition de l'utilisateur des moyens afin de

2. ÉTAPE DE CRÉATION DE L'OUTIL

modifier un mouvement de base facilement.

Pour atteindre notre objectif, plusieurs sous objectifs sont nécessaires :

- Modélisation d'un personnage virtuel paramétrable,
- Définition du contenu du fichier de mouvement,
- Adaptation du mouvement aux différentes morphologies,
- Modification interactive du mouvement.

Chaque étape représente un domaine de recherche très vaste, nous avons essayé de trouver une solution à chaque étape en prenant en considération le cadre général du projet « positionnement optimal des caméras ».

2 Étape de création de l'outil

Afin de créer l'outil, nous l'avons décomposé en sous modules permettant à à l'utilisateur de :

- Créer virtuellement la salle d'expérimentation de laboratoire,
- Créer et positionner des caméras d'une façon initiale,
- Créer un objet, lui attacher des marqueurs et ensuite lui attribuer des mouvements,
- Capturer le mouvement comme une dernière étape.

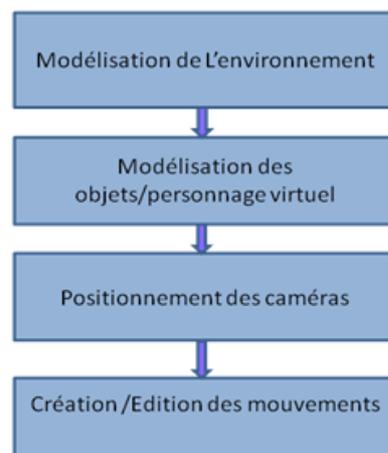


FIGURE 21 – Étape de création de l'outil

3 Architecture globale du système proposé

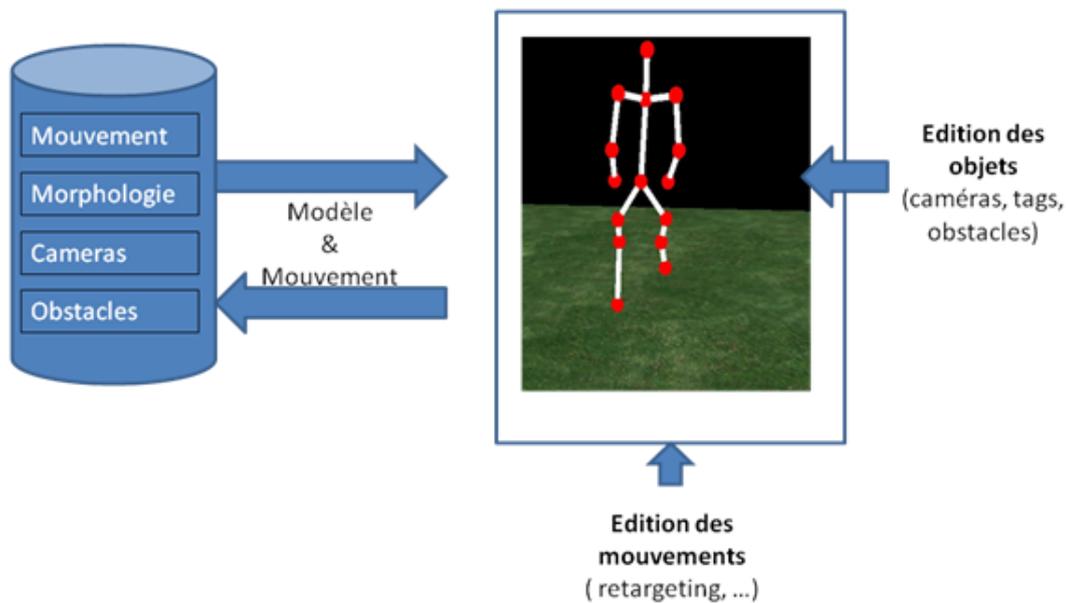


FIGURE 22 – Architecture globale du système proposé

3.1 Moyens logiciels

L'outil de modification interactive de mouvement pour le positionnement des caméras est implémenté en C++. De plus, nous utilisons la bibliothèque logicielle QT, développée en C++, qui offre des outils afin de faire notre interface graphique. La visualisation de l'environnement virtuelle et le personnage virtuel vont être délégué à une bibliothèque : Ogre3d.

3.2 Création et positionnement des caméras

La caméra est un outil de projection de l'espace 3D (l'espace de l'objet) vers l'espace image 2D. Autrement dit, cet outil nous permet de visualiser la scène 3D que nous avons créée et l'afficher dans un espace 2D.

3.2.1 Caractéristiques des caméras

Pour ajouter une caméra dans notre système, nous devons déterminer :

- La position de la caméra : c'est un vecteur 3D représentant la position $[x_{cam}, y_{cam}, z_{cam}]$ de la caméra selon les axes $[X, Y, Z]$.

3. ARCHITECTURE GLOBALE DU SYSTÈME PROPOSÉ

- L'orientation de la caméra : c'est un vecteur 3D qui représente les angles de rotation de la caméra $[\theta_x, \theta_y, \theta_z]$
- Le vecteur de direction de la caméra : c'est un vecteur 3D qui représente la direction de vue de la caméra (c'est-à-dire le point qu'elle observe)
- L'angle de champ de l'objectif qui est fixé à 45 degrés. Nous pouvons aussi indiquer les distances de near clip et de far clip, qui sont les distances minimales et maximales auxquelles doit se trouver un objet pour être affiché à l'écran.
- Représenter la caméra par un modèle physique, ce modèle est fait soit à l'aide d'un modéleur ou avec des fonctions mathématiques en prenant en compte la position et l'orientation de la caméra.

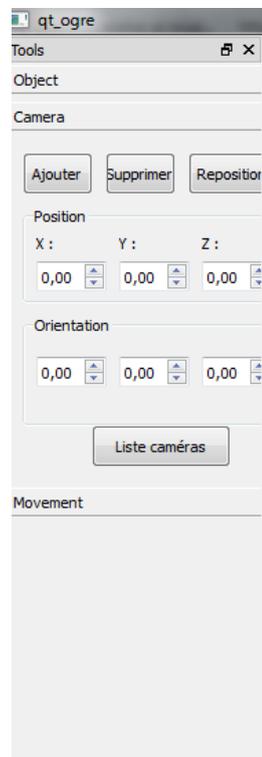


FIGURE 23 – Interface illustrant les opérations que peut effectuer l'utilisateur sur une caméra.

3.2.2 Transformations géométriques

Durant le déroulement de l'application nous pouvons déplacer la caméra selon les axes x,y,z ou les trois au même temps. Afin de se rapprocher ou de s'éloigner, nous avons ajouter une opération « zoom ». Sachant que nous pouvons aussi modifier les angles de rotations du caméra.

La figure 28 illustre la représentations des caméras dans notre outil

3. ARCHITECTURE GLOBALE DU SYSTÈME PROPOSÉ

Algorithme 7 Création des caméras

- 1: Créer une caméra et lui attribuer un nom (camera1, camera2. . . .)
 - 2: Placer la caméra dans une position choisie par l'utilisateur
 - 3: Donner une représentation physique de la caméra
 - 4: Donner le vecteur de la direction de la caméra
 - 5: Préciser l'angle de champs de l'objectif
 - 6: fin
-

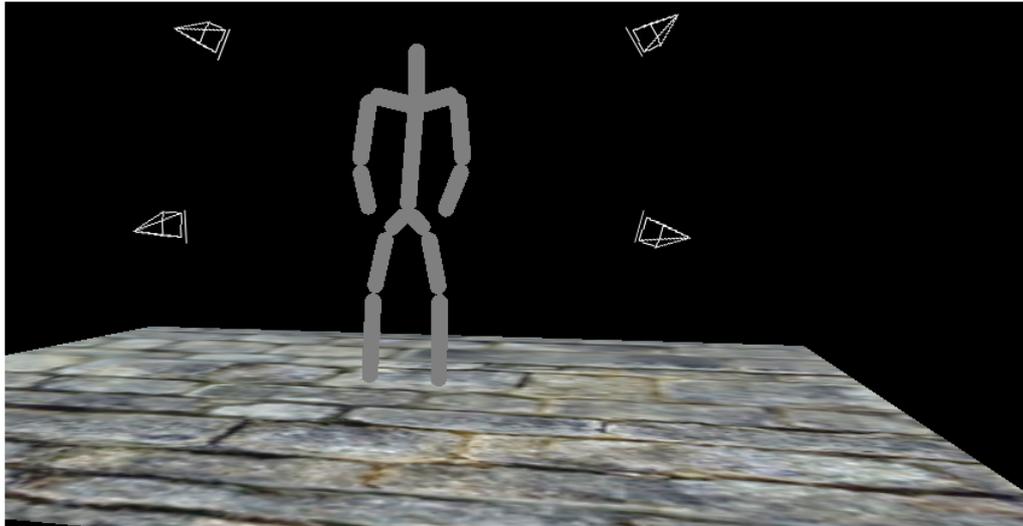


FIGURE 24 – Représentation des caméras dans le système.

3.3 Marqueurs

Dans les systèmes de capture de mouvements réels, l'acquisition des données est effectuée en utilisant des marqueurs spéciaux attachés à l'acteur. Dans le tableau 3.3 et l'algorithme 8 nous allons décrire la structure des marqueurs avec lesquels nous avons travaillé, ensuite l'algorithme montre la façon avec laquelle nous pouvons ajouter un marqueur sur le personnage virtuel.

caractéristiques	types
Numéro de marqueur	Int
Label	String
Position en x	Float
Position en y	Float
Position en z	Float
Couleur	Material

TABLE 2 – Caractéristiques des marqueurs dans notre outil

Algorithme 8 Structure des caméras

```
1: Struct marqueurs
2: debut
3: Int  $nb_{marqueur}$  ;
4: Float x,y,z ;
5: Char[10] label ;
6: fin
7: Vector <marqueur > marker ;
```

3.3.1 Ajouter un marqueur

Afin d'ajouter un marqueur nous allons activer un dispositif d'entrée/sortie « souris » en suivant les étapes ci dessous :

1. Lorsque nous cliquons sur la souris dans un point dans la scène, nous récupérons d'abord la position 2D du point où nous avons cliqué par rapport aux dimensions de la scène,
2. Nous envoyons un rayon de l'observateur vers la position 3D déjà récupérée,
3. Nous testons, si le rayon a intersecté une partie du corps du personnage ou non, s'il y a une intersection, nous calculons la position 3D du marqueur et nous l'ajoutons dans cette position, comme nous ajoutons ces informations dans le vecteur qui contient l'ensemble de marqueurs dans la scène.

Algorithme 9 Ajouter marqueur

```
1: Récupérer position 2D
2: Envoyer un rayon de l'observateur vers le point 3D
3: si (Intersection (rayon,cube))
4: Début
5: Récupérer le point 3D de l'intersection
6: Création de l'objet « sphère » représentant le marqueur et le placer dans la position point
7: Récupérer la position locale du marqueur et l'enregistrer dans un vecteur
8: fin
9: fin
```

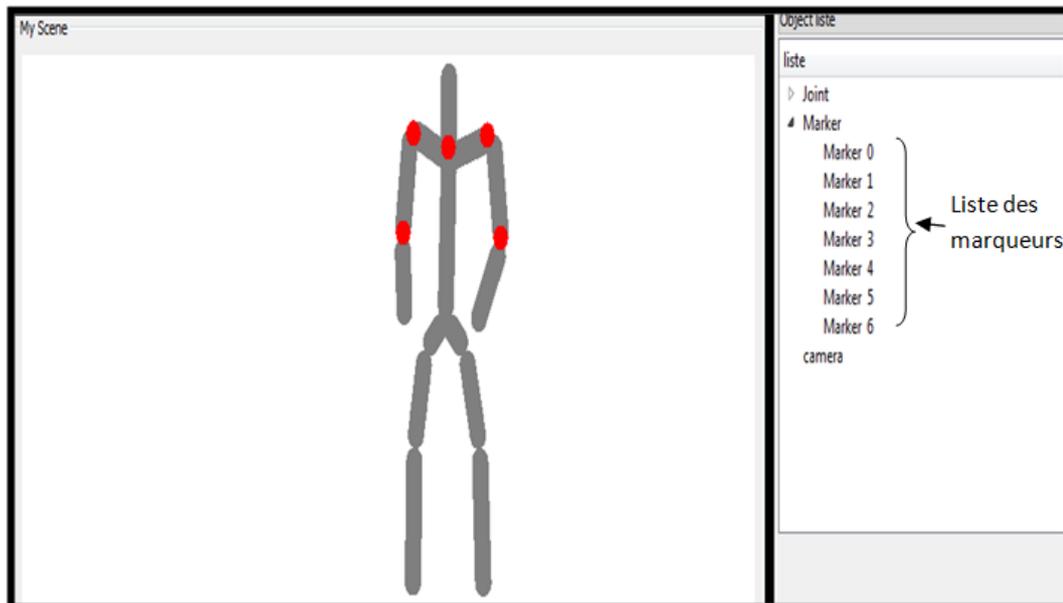


FIGURE 25 – Représentation des marqueurs

3.4 Mouvement

Les caméras virtuelles que nous avons créées vont capturer et suivre des mouvements généralement d'un personnage , Au début de notre système, la base du mouvement est vide, les utilisateurs ont la possibilité d'importer des mouvements depuis des fichier C3D et bvh, (Figure26).

Dans les sections 3.4.1 et 3.4.1, nous décrivons le module d'importation des fichiers C3D Nous décrivons le module d'importation des mouvement dans notre modèle où nous devons récupérer ces informations. Pour cela, nous avons divisé cette étape en deux sous phases :

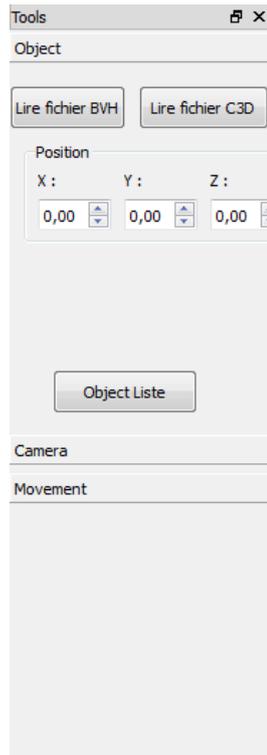


FIGURE 26 – Interface sert à importer des fichiers

3.4.1 Récupérer des informations de la section header d'un fichier C3D :

Dans cette partie nous récupérons le nombre des marqueurs, canaux analogiques et le nombre de frames.

Algorithme 10 Lecture des fichiers C3d

- 1: Ouvrir le fichier
 - 2: Récupérer le nombre des marqueurs
 - 3: Récupérer le nombre de frames
 - 4: Récupérer un pointeur vers la section paramètre
 - 5: Fermer le fichier
 - 6: Fin
-

Récupérer les informations de la partie paramètre

Dans cette section nous récupérons principalement les positions 3D de chaque marqueur dans chaque frame, en se basant sur les informations déjà récupérées dans la partie précédente.

Notons qu'il y a une autre fonction qui permet de récupérer les labels des marqueurs.

3. ARCHITECTURE GLOBALE DU SYSTÈME PROPOSÉ

Algorithme 11 Lecture des fichiers C3d -partie paramètres-

- 1: Ouvrir le fichier
 - 2: Accéder à la partie paramètre
 - 3: Pour (chaque frame)
 - 4: Début
 - 5: Pour (chaque marqueur)
 - 6: Début
 - 7: Récupérer la position 3D du marqueur
 - 8: Enregistrer la position dans un vecteur dynamique
 - 9: Fin
 - 10: Fin
 - 11: Fin
-

Nous avons testé cet algorithme, en important un fichier de mouvement d'un sujet, la figure 27 illustre les résultats.

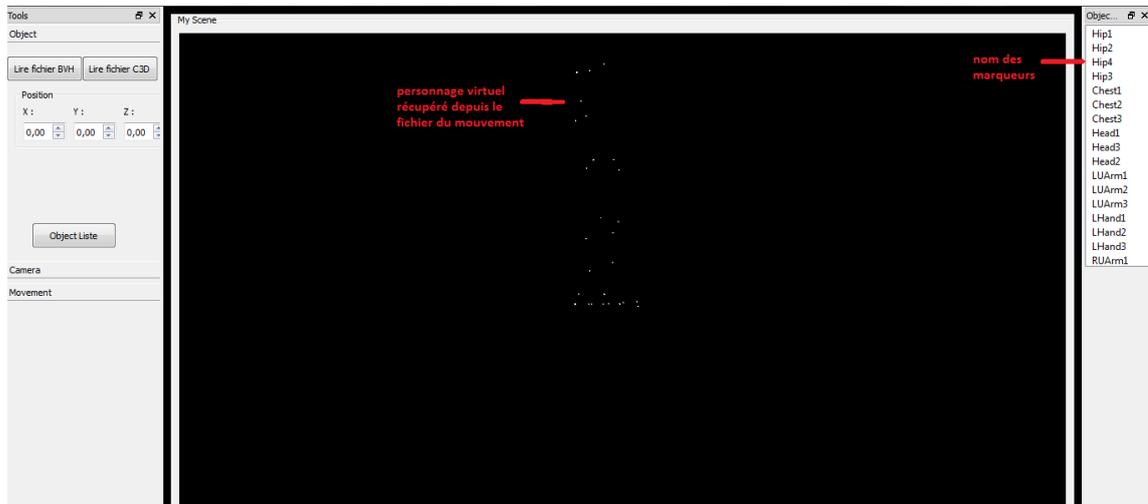


FIGURE 27 – Résultat de l'importation de fichier du mouvement : Elle représente le personnage qui est un ensemble des marqueurs copiés depuis le fichier du mouvement, ainsi les labels de ces derniers

3.5 Création de la Base de mouvement

La base de mouvement est constituée d'un ensemble de fichiers récupéré depuis les scènes de simulation.

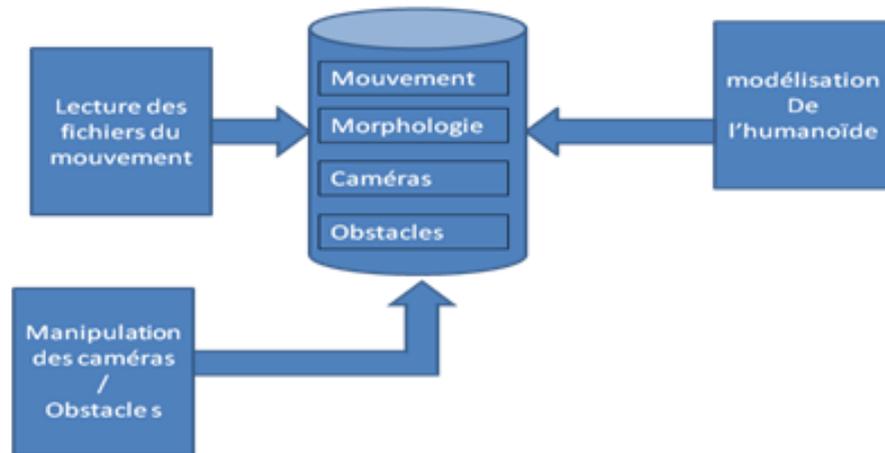


FIGURE 28 – Création de la base du mouvement

Après chaque simulation nous récupérons un fichier constitué des informations liées à la simulation : les caméras, les objets, le personnage. . . c'est le fichier intermédiaire entre les module de projet de positionnement des caméras.

3.5.1 Fichier de mouvements

Il existe plusieurs types de fichier : c3d, bvh. . . , qui sont utilisés fréquemment dans les systèmes d'animation, soit commercial comme 3dsmax , maya. , soit dans les travaux académiques de l'animation. Dans notre application, ces types de fichiers ne répondent pas à nos besoins car l'objectif essentiel du projet n'est pas l'animation mais plutôt le positionnement des caméras. Nous avons besoin d'un fichier commun entre les différents modules du projet (trois modules), un fichier qui contient toutes les informations nécessaires pour ces modules. Nous avons défini un format d'un fichier .txt qui est constitué de 6 parties :

La première partie contient les informations générales sur le mouvement comme le nombre de frames, le nom du mouvement ainsi que le nombre des caméras et le champ de vision de la caméra. La deuxième partie concerne la liste des noms des articulations du sujet, alors que la troisième partie contient les vecteurs des angles d'orientation pour chaque jointure, pour la racine, il y a un autre vecteur c'est le vecteur de translation. Dans la partie suivante, nous décrivons la hiérarchie du sujet : les relations entre les jointures et les proportions des membres du corps. L'avant dernière partie est réservée aux objets de la scène (obstacles, voiture. . .) dans cette partie, nous avons défini la géométrie de ces derniers ainsi que leurs positions. La dernière partie est réservée aux caméras, elle contient la configuration de chaque caméra, représentée par deux vecteurs : le vecteur de

la position et le vecteur de l'orientation de la caméra.

La modification de chaque partie au cours de l'exécution est facile. Par exemple, dans un premier temps, le champ du nombre de caméras est égal à zéro. Lorsqu'on crée des caméras au cours de l'exécution, on modifie ce champ dans la première partie. Les informations de la caméra créée sont ajoutées dans la dernière partie du fichier.

3.5.2 Édition du mouvement

L'édition du mouvement est le noyau de notre thèse. Dans les deux chapitres précédents, nos objectifs ont été détaillés pour effectuer cette étape. Nous avons détaillé aussi nos propositions en vue de réaliser ces derniers.

4 Conclusion

Le but de ce travail est de fournir aux utilisateurs de système de capture de mouvement VICON un outil d'aide à modifier rapidement et facilement des mouvements pré-existants en plus de créer et gérer des caméras simulant la scène prévue à capturer.

Nous avons présenté dans cet annexe l'architecture logicielle de notre outil, où nous avons créé des marqueurs, caméras et personnage virtuel afin de simuler la scène à capturer.

Bibliographie

- [1] Yeuhi Abe, C Karen Liu, and Zoran Popović. Momentum-based parameterization of dynamic character motion. *Graphical models*, 68(2) :194–211, 2006.
- [2] Brian Allen, Derek Chu, Ari Shapiro, and Petros Faloutsos. On the beat! : timing and tension for dynamic characters. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 239–247. Eurographics Association, 2007.
- [3] Kenji Amaya, Armin Bruderlin, and Tom Calvert. Emotion from motion. In *Graphics interface*, volume 96, pages 222–229. Citeseer, 1996.
- [4] Okan Arikan, David A Forsyth, and James F O’Brien. Motion synthesis from annotations. *ACM Transactions on Graphics (TOG)*, 22(3) :402–408, 2003.
- [5] Andreas Aristidou and Joan Lasenby. Fabrik : a fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5) :243–260, 2011.
- [6] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or. Action synopsis : pose selection and illustration. *ACM Transactions on Graphics (TOG)*, 24(3) :667–676, 2005.
- [7] Matthieu Aubry. *Modélisation et apprentissage de synergies pour le contrôle du mouvement de personnages virtuels-Application au geste d’atteinte de cible*. PhD thesis, Université de Bretagne occidentale-Brest, 2010.
- [8] Norman I Badler, Kamran H Manoochehri, and Graham Walters. Articulated figure positioning by multiple constraints. *Computer Graphics and Applications, IEEE*, 7(6) :28–38, 1987.
- [9] Paolo Baerlocher. *Inverse kinematics techniques for the interactive posture control of articulated figures*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2001.

- [10] Paolo Baerlocher and Ronan Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The visual computer*, 20(6) :402–417, 2004.
- [11] Douglas J Beck and DB Chaffin. An evaluation of inverse kinematics models for posture prediction. *Computer Applications in Ergonomics, Occupational safety and health*, pages 329–336, 1992.
- [12] Ahlem Bentrah, Abdelhamid Djeflal, Mc Babahenini, Christophe Gillet, Philippe Pudlo, and Abdelmalik Taleb-Ahmed. Full body adjustment using iterative inverse kinematic and body parts correlation. In *Computational Science and Its Applications–ICCSA 2014*, pages 681–694. Springer, 2014.
- [13] Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools*, 2(4) :1–13, 1997.
- [14] Rama Bindiganavale and Norman I Badler. Motion abstraction and mapping with spatial constraints. In *Modelling and Motion Capture Techniques for Virtual Environments*, pages 70–82. Springer, 1998.
- [15] Ronan Boulic, Benoît Le Calennec, Martin Herren, and Herbert Bay. Experimenting prioritized ik for motion editing. In *Eurographics 2003, Slides & Videos*, number VRLAB-CONF-2007-024, 2003.
- [16] Ronan Boulic, Branislav Ulicny, and Daniel Thalmann. Versatile walk engine. *Journal of Game Development*, 1(1) :29–52, 2004.
- [17] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 183–192. ACM Press/Addison-Wesley Publishing Co., 2000.
- [18] Joel Brown, Jean-Claude Latombe, and Kevin Montgomery. Real-time knot-tying simulation. *The Visual Computer*, 20(2-3) :165–179, 2004.
- [19] Armin Bruderlin and Lance Williams. Motion signal processing. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 97–104. ACM, 1995.
- [20] Samuel R Buss. *3D computer graphics : a mathematical introduction with OpenGL*. Cambridge University Press, 2003.
- [21] Adrian A Canutescu and Roland L Dunbrack Jr. Cyclic coordinate descent : A robotics algorithm for protein loop.

- [22] Schubert R Carvalho, Ronan Boulic, and Daniel Thalmann. Interactive low-dimensional human motion synthesis by combining motion models and pik. *Computer Animation and Virtual Worlds*, 18(4-5) :493–503, 2007.
- [23] Jinxiang Chai and Jessica K Hodgins. Performance animation from low-dimensional control signals. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 686–696. ACM, 2005.
- [24] Kwang-Jin Choi and Hyeong-Seok Ko. On-line motion retargetting. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 32–42. IEEE, 1999.
- [25] Charles E Clauser, John T McConville, and John W Young. Weight, volume, and center of mass of segments of the human body. Technical report, DTIC Document, 1969.
- [26] Michael F Cohen. Interactive spacetime control for animation. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 293–302. ACM, 1992.
- [27] Nicolas Courty and Elise Arnaud. Inverse kinematics using sequential monte carlo methods. In *Articulated Motion and Deformable Objects*, pages 1–10. Springer, 2008.
- [28] John J Craig. *Introduction to robotics : mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River, 2005.
- [29] Farid El Hafi. *Modélisation dynamique du mouvement : application à la marche et à l'enjambement d'obstacle*. PhD thesis, 2000.
- [30] AE Engin and S-M Chen. Statistical data base for the biomechanical properties of the human shoulder complex i : Kinematics of the shoulder complex. *Journal of biomechanical engineering*, 108(3) :215–221, 1986.
- [31] Andrew Feng, Yazhou Huang, Yuyu Xu, and Ari Shapiro. Automating the transfer of a generic set of behaviors onto a virtual character. In *Motion in Games*, pages 134–145. Springer, 2012.
- [32] Laetitia Fradet. *Analyse biomécanique, modélisation et simulation du tir au handball*. PhD thesis, Rennes 2, 2003.
- [33] Aurel FRATU, Jean-François BRETHÉ, and Mariana FRATU. Redundant inverse kinematics system for obstacles avoidance, 2010.
- [34] Michael Gleicher. Comparing constraint-based motion editing methods. *Graphical models*, 63(2) :107–134, 2001.

- [35] Michael Gleicher and Peter Litwinowicz. Constraint-based motion adaptation. *The journal of visualization and computer animation*, 9(2) :65–94, 1998.
- [36] F Sebastian Grassia. Practical parameterization of rotations using the exponential map. *Journal of graphics tools*, 3(3) :29–48, 1998.
- [37] Keith Grochow, Steven L Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 522–531. ACM, 2004.
- [38] H-Anim Humanoid Animation Working Group. <http://www.h-anim.org>.
- [39] 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.
- [40] Li Han and Lee Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Robotics : Science and systems*, pages 177–184, 2006.
- [41] Chris Hecker, Bernd Raabe, Ryan W Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, volume 27, page 27. ACM, 2008.
- [42] Edmond SL Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. In *ACM Transactions on Graphics (TOG)*, volume 29, page 33. ACM, 2010.
- [43] Gilles Hoffmann, Isabelle Laffont, Sylvain Hanneton, and Agnès Roby-Brami. How to extend the elbow with a weak or paralyzed triceps : control of arm kinematics for aiming in c6–c7 quadriplegic patients. *Neuroscience*, 139(2) :749–765, 2006.
- [44] Jing Huang and Catherine Pelachaud. An efficient energy transfer inverse kinematics solution. In *Motion in Games*, pages 278–289. Springer, 2012.
- [45] Leslie Ikemoto, Okan Arikan, and David Forsyth. Quick transitions with cached multi-way blends. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 145–151. ACM, 2007.
- [46] Leslie Ikemoto, Okan Arikan, and David Forsyth. Generalizing motion edits with gaussian processes. *ACM Transactions on Graphics (TOG)*, 28(1) :1, 2009.
- [47] Odest Chadwicke Jenkins and Maja J Mataric. Deriving action and behavior primitives from human motion data. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2551–2556. IEEE, 2002.

- [48] Il-Kwon Jeong, Ki Ju Park, Sung Min Baek, and Inho Lee. Implementation of a motion editing system. In *Virtual Systems and Multimedia, 2001. Proceedings. Seventh International Conference on*, pages 761–769. IEEE, 2001.
- [49] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Attention, Perception, & Psychophysics*, 14(2) :201–211, 1973.
- [50] Marcelo Kallmann. Scalable solutions for interactive virtual humans that can manipulate objects. In *AIIDE*, pages 69–75, 2005.
- [51] Marcelo Kallmann. Analytical inverse kinematics with body posture control. *Computer Animation and Virtual Worlds*, 19(2) :79–91, 2008.
- [52] Marcelo Kallmann, Amaury Aubel, Tolga Abaci, and Daniel Thalmann. Planning collision-free reaching motions for interactive object manipulation and grasping. In *Computer Graphics Forum*, volume 22, pages 313–322. Wiley Online Library, 2003.
- [53] Erika Karabinova Makk. *Modélisation du membre inférieur à des fins ergonomiques par l'exploitation de marqueurs externes*. PhD thesis, 1999.
- [54] Ben Kenwright. Responsive biped character stepping : When push comes to shove. In *Cyberworlds (CW), 2012 International Conference on*, pages 151–156. IEEE, 2012.
- [55] Ben Kenwright, Richard Davison, and Graham Morgan. Dynamic balancing and walking for real-time 3d characters. In *Motion in Games*, pages 63–73. Springer, 2011.
- [56] Charles A Klein and C-H Huang. Review of pseudoinverse control for use with kinematically redundant manipulators. *Systems, Man and Cybernetics, IEEE Transactions on*, (2) :245–250, 1983.
- [57] Taku Komura, Atsushi Kuroda, Shunsuke Kudoh, Tai Chiew Lan, and Yoshihisa Shinagawa. An inverse kinematics method for 3d figures with motion data. In *Computer Graphics International, 2003. Proceedings*, pages 266–271. IEEE, 2003.
- [58] Taku Komura, Yoshihisa Shinagawa, and TL Kunii. Attaching physiological effects to motion-captured data. In *Graphics Interface*, pages 27–36, 2001.
- [59] James U Korein. *A geometric investigation of reach*. MIT press, 1986.
- [60] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM transactions on graphics (TOG)*, volume 21, pages 473–482. ACM, 2002.

- [61] Lucas Kovar, John Schreiner, and Michael Gleicher. Footskate cleanup for motion capture editing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 97–104. ACM, 2002.
- [62] R Kulpa, F Multon, and Bruno Arnaldi. Morphology-independent representation of motions for interactive human-like animation. In *Computer Graphics Forum*, volume 24, pages 343–351. Wiley Online Library, 2005.
- [63] John Lasseter. Principles of traditional animation applied to 3d computer animation. In *ACM Siggraph Computer Graphics*, volume 21, pages 35–44. ACM, 1987.
- [64] Benoît Le Calennec and Ronan Boulic. Interactive motion deformation with prioritized constraints. *Graphical Models*, 68(2) :175–193, 2006.
- [65] Benoît Le Calennec and Ronan Boulic. Robust kinematic constraint detection for motion data. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 281–290. Eurographics Association, 2006.
- [66] Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 491–500. ACM, 2002.
- [67] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 39–48. ACM Press/Addison-Wesley Publishing Co., 1999.
- [68] Yan Li, Tianshu Wang, and Heung-Yeung Shum. Motion texture : a two-level statistical model for character motion synthesis. In *ACM Transactions on Graphics (ToG)*, volume 21, pages 465–472. ACM, 2002.
- [69] C Karen Liu and Zoran Popović. Synthesis of complex dynamic character motion from simple animations. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 408–416. ACM, 2002.
- [70] Huajun Liu, Xiaolin Wei, Jinxiang Chai, Inwoo Ha, and Taehyun Rhee. Realtime human motion control with a small number of inertial sensors. In *Symposium on Interactive 3D Graphics and Games*, pages 133–140. ACM, 2011.
- [71] Li Liu, Zhao-qi Liu, Zhu Deng-ming, and Shi-Hong Xia. Motion edit with collision avoidance. 2006.

- [72] Anthony A Maciejewski and Charles A Klein. Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The international journal of robotics research*, 4(3) :109–117, 1985.
- [73] Stéphane Ménardais, Richard Kulpa, Franck Multon, and Bruno Arnaldi. Synchronization for dynamic blending of motions. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 325–335. Eurographics Association, 2004.
- [74] Jianyuan Min, Yen-Lin Chen, and Jinxiang Chai. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics (TOG)*, 29(1) :9, 2009.
- [75] Jean-Sébastien Monzani, Paolo Baerlocher, Ronan Boulic, and Daniel Thalmann. Using an intermediate skeleton and inverse kinematics for motion retargeting. In *Computer Graphics Forum*, volume 19, pages 11–19. Wiley Online Library, 2000.
- [76] R Muller-Cajar and R Mukundan. Triangulation—a new algorithm for inverse kinematics. 2007.
- [77] Franck Multon. *Contrôle du mouvement des humanoïdes de synthèse*. PhD thesis, 1998.
- [78] Franck Multon. *Analyse, modélisation et simulation du mouvement humain*. PhD thesis, Université Rennes 1, 2006.
- [79] Yoshihiko Nakamura and Hideo Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of dynamic systems, measurement, and control*, 108(3) :163–171, 1986.
- [80] Naturalmotion. <http://www.naturalmotion.com/>.
- [81] Michael Neff and Eugene Fiume. Aer : aesthetic exploration and refinement for expressive character animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 161–170. ACM, 2005.
- [82] Michael Neff and Yejin Kim. Interactive editing of motion style using drives and correlations. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. ACM, 2009.
- [83] Eng-Jon Ong and Adrian Hilton. Learnt inverse kinematics for animation synthesis. *Graphical Models*, 68(5) :472–483, 2006.

- [84] Richard P Paul. *Robot manipulators : mathematics, programming, and control : the computer control of robot manipulators*. Richard Paul, 1981.
- [85] Daniel Raunhardt and Ronan Boulic. Motion constraint. *The Visual Computer*, 25(5-7) :509–518, 2009.
- [86] Charles Rose, Michael F Cohen, and Bobby Bodenheimer. Verbs and adverbs : Multidimensional motion interpolation. *Computer Graphics and Applications, IEEE*, 18(5) :32–40, 1998.
- [87] Charles Rose, Brian Guenter, Bobby Bodenheimer, and Michael F Cohen. Efficient generation of motion transitions using spacetime constraints. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 147–154. ACM, 1996.
- [88] Charles F Rose III, Peter-Pike J Sloan, and Michael F Cohen. Artist-directed inverse-kinematics using radial basis function interpolation. In *Computer Graphics Forum*, volume 20, pages 239–250. Wiley Online Library, 2001.
- [89] Alla Safonova and Jessica K Hodgins. Construction and optimal search of interpolated motion graphs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 106. ACM, 2007.
- [90] Alla Safonova, Jessica K Hodgins, and Nancy S Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 514–521. ACM, 2004.
- [91] Ari Shapiro, Petros Faloutsos, and Victor Ng-Thow-Hing. Dynamic animation and control environment. In *Proceedings of graphics interface 2005*, pages 61–70. Canadian Human-Computer Communications Society, 2005.
- [92] Hyun Joon Shin, Lucas Kovar, and Michael Gleicher. Physical touch-up of human motions. In *Pacific Conference on Computer Graphics and Applications*, pages 194–203, 2003.
- [93] Hyun Joon Shin, Jehee Lee, Sung Yong Shin, and Michael Gleicher. Computer puppetry : An importance-based approach. *ACM Transactions on Graphics (TOG)*, 20(2) :67–94, 2001.
- [94] Hubert Shum and Edmond SL Ho. Real-time physical modelling of character movements with microsoft kinect. In *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, pages 17–24. ACM, 2012.

- [95] Bruno Siciliano and J-JE Slotine. A general framework for managing multiple tasks in highly redundant robotic systems. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1211–1216. IEEE, 1991.
- [96] Kwang Won Sok, Katsu Yamane, Jehee Lee, and Jessica Hodgins. Editing dynamic human motions via momentum and force. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 11–20. Eurographics Association, 2010.
- [97] Sandra Sudarsky and Donald House. Motion capture data manipulation and reuse via b-splines. In *Modelling and Motion Capture Techniques for Virtual Environments*, pages 55–69. Springer, 1998.
- [98] Seyoon Tak, Oh-young Song, and Hyeong-Seok Ko. Spacetime sweeping : An interactive dynamic constraints solver. In *Computer Animation, 2002. Proceedings of*, pages 261–270. IEEE, 2002.
- [99] Deepak Tolani, Ambarish Goswami, and Norman I Badler. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical models*, 62(5) :353–388, 2000.
- [100] Maxime Tournier, Xiaomao Wu, Nicolas Courty, Elise Arnaud, and Lionel Reveret. Motion compression using principal geodesics analysis. In *Computer Graphics Forum*, volume 28, pages 355–364. Wiley Online Library, 2009.
- [101] Munetoshi Unuma, Ken Anjyo, and Ryoza Takeuchi. Fourier principles for emotion-based human figure animation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 91–96. ACM, 1995.
- [102] Munetoshi Unuma and Ryoza Takeuchi. Generation of human motion with emotion. In *Computer Animation 91*, pages 77–88. Springer, 1991.
- [103] Luis Unzueta, Manuel Peinado, Ronan Boulic, and Ángel Suescun. Full-body performance animation with sequential inverse kinematics. *Graphical models*, 70(5) :87–104, 2008.
- [104] JP Verriest, MA Rezgui, and X Wang. Experimental validation of arm reach movement simulation. In *Proc., 12th Congress of International Ergonomics Association*, pages 342–344, 1994.

- [105] L-CT Wang and Chih-Cheng Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *Robotics and Automation, IEEE Transactions on*, 7(4) :489–499, 1991.
- [106] Xuguang Wang. A behavior-based inverse kinematics algorithm to predict arm prehension postures for computer-aided ergonomic evaluation. *Journal of biomechanics*, 32(5) :453–460, 1999.
- [107] Xuguang Wang and Jean Pierre Verriest. A geometric algorithm to predict the arm reach posture for computer-aided ergonomic evaluation. *The journal of visualization and computer animation*, 9(1) :33–47, 1998.
- [108] Xiaolin K Wei and Jinxiang Chai. Intuitive interactive human-character posing with millions of example poses. *Computer Graphics and Applications, IEEE*, 31(4) :78–88, 2011.
- [109] Chris Welman. *Inverse kinematics and geometric constraints for articulated figure manipulation*. PhD thesis, Simon Fraser University, 1993.
- [110] Daniel E Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems*, 1969.
- [111] Andrew Witkin and Michael Kass. Spacetime constraints. In *ACM Siggraph Computer Graphics*, volume 22, pages 159–168. ACM, 1988.
- [112] Andrew Witkin and Zoran Popovic. Motion warping. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 105–108. ACM, 1995.
- [113] William A Wolovich and H Elliott. A computational technique for inverse kinematics. In *Decision and Control, 1984. The 23rd IEEE Conference on*, pages 1359–1363. IEEE, 1984.
- [114] Benjun Wu, Clinton P Conrad, Arnauld Heuret, Carolina Lithgow-Bertelloni, and Serge Lallemand. Reconciling strong slab pull and weak plate bending : The plate motion constraint on the strength of mantle slabs. *Earth and Planetary Science Letters*, 272(1) :412–421, 2008.
- [115] Katsu Yamane and Yoshihiko Nakamura. Dynamics filter-concept and implementation of online motion generator for human figures. *Robotics and Automation, IEEE Transactions on*, 19(3) :421–432, 2003.

- [116] Jianmin Zhao and Norman I Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics (TOG)*, 13(4) :313–336, 1994.
- [117] Victor Brian Zordan and Jessica K Hodgins. Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 89–96. ACM, 2002.
- [118] Victor Brian Zordan, Anna Majkowska, Bill Chiu, and Matthew Fast. Dynamic response for motion capture animation. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 697–701. ACM, 2005.
- [119] Victor Brian Zordan and Nicholas C Van Der Horst. Mapping optical motion capture data to skeletal motion using a physical model. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 245–250. Eurographics Association, 2003.