Paramétrage et Capture Multicaméras du Mouvement Humain

David Knossow

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

THÈSE
pour obtenir le grade de
DOCTEUR DE L'INPG

Spécialité : "Imagerie, Vision, Robotique"

préparée au laboratoire Gravir - Imag - Inria Rhône-Alpes
dans le cadre de **l'Ecole Doctorale *"Mathématiques, Sciences et Technologies de l'Information, Informatique"***

présentée et soutenue publiquement par

David Knossow
Le 23 Avril 2007

Titre :

ANALYSE ET CAPTURE MULTI-CAMÉRAS DU MOUVEMENT HUMAIN

Directeurs de Thèse :
Radu Horaud, Rémi Ronfard

JURY

M.	**James L. Crowley**	Président
M.	**Ian Reid**	Rapporteur
M.	**Nikos Paragios**	Rapporteur
M.	**Adrian Hilton**	Examinateur
M.	**Luc Robert**	Membre invité
M.	**Radu Horaud**	Directeur de thèse
M.	**Rémi Ronfard**	Co-directeur de thèse

Quand un bon sculpteur modèle des corps humains,
il ne représente pas seulement la musculature,
mais aussi la vie qui les réchauffe.
A. Rodin

A : C., qui a su être si patiente...
M.K.
M.K.
S.K.
M.K.

Remerciements

Je remercie tout d'abord Radu et Rémi d'avoir bien voulu m'accepter comme doctorant au sein de l'équipe PERCEPTION. Ils ont su m'encadrer et me permettre de mener à bien tous mes travaux de thèse.

Je remercie ensuite tous les membres du jury d'avoir accepté que je présente mes travaux de thèse et de m'avoir accordé le grade de docteur.

Tout au long de ces trois années passées dans l'équipe PERCEPTION, j'ai pu apprécier la collaboration avec Loic Lefort. Sa contribution très forte m'a permis d'aboutir dans mes travaux. J'aimerais aussi remercier Joost Van de Weijer, Frédéric Devernay et Pierre Brice Wieber pour toutes les échanges constructifs que nous avons pu avoir et qui m'ont permis d'apporter des réponses aux difficultés rencontrées. De même, je tiens à remercier les membres du projet SEMOCAP, projet sans lequel la thèse ne se serait pas déroulée de la même manière.

Je pense aussi à Peter S., Vincent L., Stephane R., Xavier D., Alba et Pao, Markus M., Florian G., Bertrand H., Alexander K., Bill T., Ankur A., Navneet D., Matis D., Elise A., Elise T., Aude J., Srikumar R., Jean Philippe T., Andrei Z., Kiran V. pour l'ambiance et la bonne humeur.

J'aimerais enfin remercier le club robot de m'avoir permis de faire parti d'une aventure palpitante.

Résumé

Au cours de cette thèse, nous avons abordé la problématique du suivi du mouvement humain. De manière générale, le suivi du mouvement à l'aide de marqueurs (optiques, magnétiques, ...) est très étudié que ce soit pour des applications médicales ou l'aide à l'animation 3D. Dans le cadre de la thèse, nous proposons une méthode alternative à ces systèmes coûteux et souvent contraignants. Nous proposons une approche utilisant plusieurs caméras vidéos, permettant de s'affranchir de contraintes sur l'environnement de capture ainsi que de la pose de marqueurs sur l'acteur. L'absence de marqueurs complexifie la recherche d'une information pertinente dans les images mais aussi la corrélation de cette information entre les différentes images (problèmes de traitement d'image et de suivi). De plus, il est difficile de d'interpréter cette information en terme de mouvements articulaires du corps humain (problèmes de géométrie).

Dans cette thèse, nous nous intéressons à une approche utilisant les contours occultants des différentes parties du corps. Nous avons étudié le lien entre le mouvement articulaire du corps humain et le mouvement apparent des contours vus par une caméra. Nous avons également étudié plusieurs stratégies pour l'extraction des contours occultants. Cette approche nous donne un grand nombre d'observations, que nous réalisons avec un petit nombre de caméras (4 à 6). Une minimisation de l'erreur entre les contours extraits des images et la projection du modèle sur chacune de ces images nous permet d'estimer le mouvement de l'acteur filmé.

Grâce à la plate-forme GrImage mise en place au sein de l'INRIA Rhône-Alpes, nous avons pu effectuer plusieurs expérimentations permettant de valider l'approche proposée. De plus, une collaboration avec l Université de Rennes nous a permis de mettre en place des expérimentations permettant de comparer qualitativement notre méthode aux résultats du système VICON généralement utilisé pour l'animation 3D.

Parmi les perspectives ouvertes par notre travail, nous montrons que l'utilisation d'images video permet entre autres de produire des informations complémentaires telles que les contacts entre les parties du corps de l'acteur ou entre l'acteur et son environnement. Ces informations sont particulièrement importantes pour la ré-utilisation des mouvements en animation 3D.

Abstract

This manuscript deals with the problem of markerless human motion capture. Generally talking, marker based human motion capture have been thoroughly studied for medical use or to provide help for 3D character animation (cinematography or video games). This Ph. D. allowed us to study an alternative to those expensive and constraining systems. We propose an approach that relies on the use of multiple cameras and that avoids most of the constraints on the environment and the use of markers to perform the motion capture. The absence of markers makes harder the problem of extracting relevant information from images but also to correlate this information between images (image processing and tracking issues). Moreover, interpreting this extracted information in terms of joint parameters motion is not an easy task (geometry issues).

We propose an approach that relies on occluding contours of the human body. We studied the link between motion parameters and the appearant motion of the edges in images. We also studied multiple strategies to extract edges from images. This approach allows to get many information from a rather small number of cameras (4 to 6). Minimizing the error between the extracted edges and the projection of the 3D model onto the images allows to estimate the motion parameters of the actor.

Thanks to the GrImage platform, we lead experiments to validate the proposed method. Moreover, we had the opportunity to qualitatively compare our results with those obtained using a widely used motion capture system (VICON). We obtained very promising results. Among the opened issues, we show that using video based motion capture allows to provide additional hints such as contacts between body parts or between the actor and its environment. This information is particularly relevant for improving character animation.

Table des matières

Table des figures

Chapitre 1

Introduction générale

Sommaire

Introduction à la thèse

A l'origine de cette thèse, il y a un projet industriel porté par une PME (ARTEFACTO) pour rendre la capture du mouvement accessible aux petits studios d'animation *3D*.

Imaginez-vous un jeune graphiste embauché par une société de production de jeux vidéo éducatifs. Sa nouvelle société est une société dynamique où les idées de jeux innovants, ludiques et pédagogiques ne manquent pas. Il va travailler sur un nouveau projet mettant en scène des enfants partant à la conquête du savoir. Son travail dans ce projet : animer les personnages du jeu. Etant donné les jeux actuels et le besoin de réalisme qu'ont les enfants d'aujourd'hui, sa tâche sera de faire des animations les plus réalistes possibles. Il est motivé, cependant, il se prend à penser aux grands éditeurs de jeux ou de films qui utilisent des systèmes de capture de mouvement pour faciliter le travail des animateurs et surtout baisser les temps de production[1]. Et pourquoi pas lui, pourquoi son entreprise n'en n'aurait pas ? La réponse semble assez simple : le prix de ces systèmes. Le système le plus couramment utilisé, VICON, coûte de l'ordre de 350 000 € et pour une petite entreprise, c'est cher. La location est une idée, mais avoir son propre système c'est mieux. Cependant, les moyens de la société sont limités. L'objectif est donc de réfléchir à un système de capture de mouvement financièrement accessible et flexible. D'une part le matériel doit être standard et réutilisable à d'autres fins. D'autre part, le système doit être transportable, facilement démontable et fiable pour faire de la pré-production de jeux. En ce qui concerne la partie technique du défi, les choix semblent évidents :

- Pour capturer l'acteur : des caméras qui filment et enregistrent des séquences vidéo. La définition peut être standard, mais le format de sortie non compressé pour garder une qualité optimale des séquences vidéo. Les caméras doivent, de plus, pouvoir être synchronisées.
- Pour effectuer les traitements : une station de travail performante.

Encore faut-il trouver comment utiliser les séquences vidéos pour aider à l'animation des personnages.

C'est ce dernier point que nous nous proposons d'aborder dans cette thèse. Plus précisément, nous allons aborder la problématique de la capture du mouvement avec pour objectif à terme de créer des bases de données de mouvement pour l'animation.

1.1 Motivation et objectifs

Dans ce manuscrit, nous allons aborder l'ensemble du processus de la capture du mouvement tout en appuyant sur la partie scientifique du problème : comment estimer le mouvement d'un acteur à partir de séquences vidéo acquises dans un environnement peu contrôlé. Le résultat de l'estimation doit être sous la forme d'un fichier exploitable par les graphistes pour animer de nouveaux personnages.

[1] Notons que PIXAR estime que le mouvement obtenu par les animateurs est nettement meilleur que celui obtenu à partir de séances de capture du mouvement. Ce studio n'utilise donc pas de système de capture du mouvement.

La contrainte principale du système est probablement l'absence de contraintes particulières pour le lieu d'acquisition ainsi que pour la tenue vestimentaire de l'acteur. De plus, le nombre de caméras doit pouvoir être variable et minimal pour une séquence de mouvement donnée (3 à 6 caméras pour des mouvements simples à complexes).

Ce problème de capture du mouvement est connu sous le nom de « capture du mouvement multi-caméras sans marqueurs ». Il s'agit d'un problème qui fait l'objet de nombreux travaux de recherche depuis le début des années 80. Sur la base de travaux précurseurs comme ceux de [97], [75] ou encore [124], les systèmes se sont développés pour intégrer des systèmes sophistiqués d'apprentissage [2], ou encore la multiplicité des flux vidéos analysés simultanément [52]. Nous pouvons maintenant concevoir des systèmes complexes avec plusieurs caméras et capables d'effectuer des traitements en temps réel [4]. Plusieurs états de l'art sur le suivi et la reconnaissance du mouvement ([3], [53], [105]) montrent à quel point les développements se sont accélérés ces dix ou quinze dernières années. De plus, deux numéros spéciaux de *Computer Vision and Image Understanding* ([72] et [73]) sont consacrés aux problèmes se rapportant à la capture du mouvement sans marqueurs. Les efforts actuels sont portés sur l'aspect interactif et temps réel de la capture du mouvement. Cependant, il existe un compromis entre l'aspect temps réel et la précision avec laquelle la capture du mouvement doit être effectuée.

Le projet dans lequel s'inscrit cette thèse a pour but de créer des bases de données de mouvement adaptables à des personnages animés. Nous nous sommes donnés comme objectif de proposer une approche rapide mais non temps réel pour le traitement des données de capture. Cette absence de contrainte de vitesse d'exécution ouvre un large champ de possibilités pour traiter les images et effectuer le suivi du mouvement de manière précise. En pratique, l'approche que nous proposons permet de traiter une minute de mouvement en deux heures de temps machine.

Dans la suite de ce chapitre, nous allons présenter le cadre de la réalisation de la thèse. Puis, nous évoquerons le déroulement de celle-ci. Nous aborderons enfin l'organisation du document.

1.2 Cadre et contexte de la thèse

Ma thèse s'est déroulée dans le cadre d'un projet national de type RIAM et financé par le CNC[2]. Le projet SEMOCAP a été une collaboration entre deux laboratoires (l'INRIA avec l'équipe MOVI et l'UHB [3] avec le LBPEM [4]) et deux industriels (ARTEFACTO, société d'architecture et de production de jeux pédagogiques et ASICA, société française spécialisée dans l'électronique). Chaque entité a apporté ses compétences :

- ARTEFACTO a été à l'initiative du projet et a proposé sa compétence dans l'animation. Elle a été coordinatrice de l'ensemble du projet et a permis la cohérence et l'avancée de l'ensemble des partenaires.

[2]Centre National de la Cinématographie
[3]Université Rennes 2 - Haute Bretagne
[4]Laboratoire de Physiologie et de Biomécanique de l'Exercice Musculaire

- ASICA a construit un système matériel de capture portable basé sur l'architecture de la plate-forme GRIMAGE de l'INRIA et sur les logiciels de la suite MVSTUDIO développée à l'INRIA.
- L'UHB avec Richard Kulpa, Franck Multon et Armel Crétual a apporté la compétence biomécanique au problème. Le fichier de mouvement issu de la capture du mouvement est spécifique à l'acteur et peut comporter des anomalies (rotations de 180° de certains membres, par exemple) dans le mouvement. Ils proposent donc de modifier le fichier de capture pour pouvoir l'adapter à toute morphologie humaine possible et de corriger les anomalies pour obtenir une animation visuellement correcte.
- L'équipe PERCEPTION (anciennement MOVI) a apporté sa compétence en vision multi-caméras. C'est au sein de cette équipe, avec Radu Horaud, Rémi Ronfard que j'ai effectué ma thèse. Dans le cadre du projet SEMOCAP j'ai travaillé avec Loïc Lefort. Nous avons proposé la méthode de suivi du mouvement : du traitement des données vidéo jusqu'à la génération d'un fichier décrivant le mouvement de l'acteur.

Au sein de l'équipe, nous avons élaboré les outils nécessaires pour la capture du mouvement, dont les différentes étapes sont décrites sur la figure 1.1. Loïc Lefort s'est occupé du calibrage des caméras, de la soustraction de fond et enfin de la méthode de dimensionnement de l'acteur (capture de l'acteur). L'étape du suivi du mouvement a été le sujet principal de ma thèse. Cependant, pour réaliser cette étape, j'ai dû aborder les problèmes du choix de la méthode, du choix du modèle *3D*, des techniques à mettre en place pour traiter les images et en extraire les données utiles à la capture du mouvement. Enfin, en raison de ce contexte industriel du projet, j'ai eu la chance de voir les algorithmes utilisés au cours de ma thèse appliqués dans un contexte d'utilisation réel et intégrés dans une application distribuée aux partenaires du projet (MVPOSER).

1.3 Déroulement de la thèse

Au cours de ces trois années de thèse, j'ai abordé différentes problématiques aussi bien d'ordre scientifique (la modélisation cinématique du corps humain, la minimisation de fonctions non linéaires, la détection de contours, le filtrage particulaire, la détection de collisions) que d'ordre pratique (utilisation avancée des cartes graphiques, création d'interfaces graphiques).

Ma thèse s'est déroulée de la manière suivante :

- En 2004, nous abordions la capture du mouvement multi-caméras pour la première fois dans notre laboratoire. Des travaux sur des approches mono-caméras ont été menés avec notamment [135] et l'ensemble des travaux présentés dans [143]. J'ai donc effectué une étude bibliographique complète sur la capture du mouvement multi-caméras. Cette étude m'a permis de comprendre les problèmatiques posées par la capture du mouvement et de mettre en place les premiers développements nécessaires pour effectuer de la capture du mouvement multi-caméras sans marqueurs. Nous avons donc défini la première ébauche de l'algorithme de capture

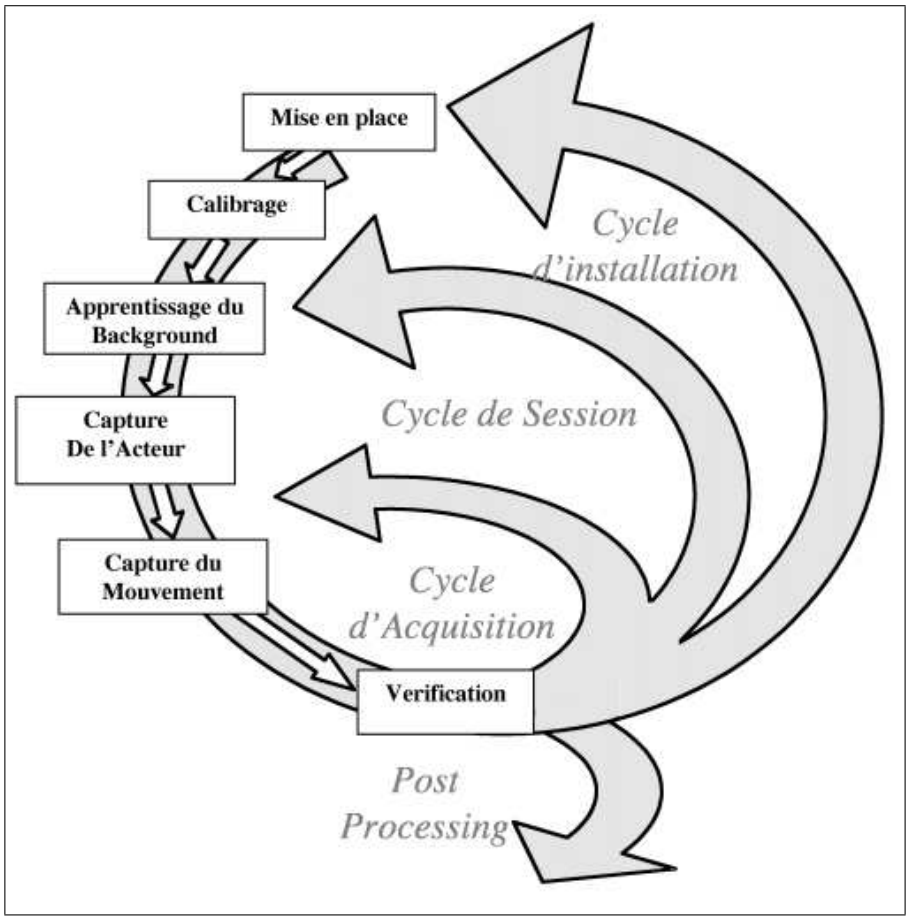


FIG. 1.1: Plusieurs étapes sont nécessaires pour effectuer la capture du mouvement.

du mouvement. Nous avons créé les outils nécessaires pour effectuer le traitement de plusieurs flux vidéo de manière simultanée. Nous avons aussi établi les développements mathématiques (comme la modélisation cinématique du squelette, la formulation du déplacement de l'observation du modèle dans les images lorsque celui-ci se déplace en *3D*, le calcul des fonctions d'erreurs ainsi que de leur jacobien pour effectuer le suivi, *etc.*). Les premiers résultats ont été obtenus en fin de première année. Les premiers tests ont portés sur des séquences d'images générées avec le logiciel POSER[5].

- En 2005, j'ai pu mettre en oeuvre une solution complète. J'ai commencé à tester les algorithmes de capture du mouvement sur des séquences d'images acquises avec la plate-forme GRIMAGE mise en place à l'INRIA au cours de l'année 2004. Nous avons pu prendre la mesure de la difficulté à utiliser des séquences réelles et les résultats sur ce type de séquences ont été obtenus au milieu de l'année et publiés en Janvier 2006 lors de la conférence ACCV ([85]). Ensuite, nous avons commencé à nous intéresser à la capture du mouvement dans des environnements peu contrôlés. Ces travaux, avec ceux de la première année, constituent la première partie de ce document de thèse. En fin de seconde année, nous avons effectué les premières acquisitions vidéo avec les partenaires du projet SEMOCAP à Rennes. Au cours de ces acquisitions nous avons pu travailler pour la première fois avec un système VICON. Ce dernier nous a permis d'effectuer une séance de capture du mouvement avec la possibilité de se comparer au système actuellement le plus utilisé.
- En 2006, je me suis attaché à faire évoluer évoluer le premier système en intégrant des contraintes physiques (c.f. chapitre 7) ainsi qu'un suivi spécifique des pieds, des mains et de la tête (c.f. chapitre 8). Nous avons aussi cherché à rendre plus robuste le suivi du mouvement. Nous aborderons ces points dans la seconde partie de ma thèse. Nous avons publié un article sur la méthode de suivi utilisant une technique adéquate de détection de contours lors du Workshop on Dynamical Vision associé à ECCV 2006 ([86]).

1.4 Plan de la thèse

Dans un premier chapitre, nous aborderons un état de l'art général sur la capture du mouvement. Nous présenterons les systèmes industriels les plus utilisés pour effectuer de la capture du mouvement. Nous nous attarderons sur la présentation d'un système de capture du mouvement avec marqueurs (VICON [6], BTS SMART-E[7], ISADORA[8], /etc). Nous établirons alors un parallèle entre les systèmes industriels et les systèmes sans marqueurs. Nous expliciterons chacune des difficultés et ferons référence aux travaux ayant abordés ces points dans le passé.

[5]Curious Labs
[6]http ://www.vicon.com/
[7]http ://www.bts.it/eng/proser/elisma.htm
[8]http ://www.troikatronix.com/isadora.html

La suite de la thèse est organisée en deux parties. La première partie présente le système de base pour la capture du mouvement, tandis que la seconde partie présente la mise en oeuvre expérimentale, les résultats ainsi que plusieurs extensions réalisées au cours de la troisième année.

Dans le premier chapitre, nous expliciterons la modélisation et le paramétrage du squelette humain. Pour cela, nous commencerons par faire des rappels de cinématique au cours desquels nous aborderons la modélisation d'une chaîne cinématique. Nous développerons dans ce même chapitre le formalisme mis en place pour modéliser le mouvement d'un point attaché à une chaîne cinématique en fonction des paramètres articulaires de celle-ci (Chapitre 3). Ce premier chapitre permettra de poser les bases mathématiques nécessaires pour effectuer le suivi. Dans le chapitre 4, nous développerons la modélisation *3D* du corps humain. Dans un premier temps nous appliquerons les développements du chapitre 3 au cas du corps humain. Puis nous développerons le modèle géométrique adopté pour modéliser l'acteur. Enfin nous donnerons la modélisation analytique du mouvement des contours du modèle *3D* observé dans les images en fonction des paramètres de pose de la chaîne cinématique, ce qui constituera la première contribution majeure de cette thèse.

Le chapitre 5, est consacré à la mise en correspondance du modèle *3D* avec les observations dans les images. Nous proposerons une approche pour l'initialisation de la capture du mouvement, avec la phase de dimensionnement du modèle *3D* ainsi que la phase d'estimation de la pose de l'acteur dans la première image de la séquence vidéo. Dans un second temps, nous développerons l'approche choisie pour estimer le mouvement de l'acteur. Pour chacune de ces étapes, nous introduirons des méthodes spécifiques de mise en correspondance. Cependant, nous montrerons que la réalisation de chacune de ces étapes n'est rien d'autre que la minimisation d'une fonction de coût dépendant de manière explicite des paramètres articulaires et dimensionnels du modèle *3D*. Nous achèverons ce chapitre en présentant la méthode de minimisation mise en place et qui a l'avantage d'être commune au trois étapes de la capture.

Dans la seconde partie de cette thèse, nous commençons par décrire la mise en oeuvre expérimentale du système de capture que nous avons réalisé dans le cadre du projet SEMOCAP ainsi que les résultats obtenus.

L'estimation du mouvement telle que nous la présenterons dans le chapitre 5 n'intègre aucune contrainte sur le déplacement du modèle. Dans les chapitres 7 et 8 nous montrerons que l'absence de contraintes lors de la minimisation peut entraîner un échec du suivi du mouvement qui nécessite de nombreuses interventions manuelles. Pour éviter des pertes du suivi fréquentes, nous proposons deux contraintes dans la minimisation : l'absence de collisions entre les différentes parties du modèle *3D* (chapitre 7) et une méthode de suivi spécifique pour les pieds, les mains et la tête permettant de rendre plus robuste le suivi de ces membres (chapitre 8).

Nous concluons en proposant quelques perspectives ouvertes par notre travail (chapitre 9).

Chapitre 2

Etat de l'art en capture de mouvement

Sommaire

Introduction au chapitre

Lorsque nous regardons autour de nous, quelle est la chose qui attire le plus notre regard ? De manière générale, nous sommes attirés par le mouvement et la couleur. Cependant, nous aurons tendance, par instinct, à évaluer un objet en mouvement plus qu'un objet statique. Que ce soit un lion féroce prêt à nous attaquer ou encore une voiture qui arrive trop vite... Le mouvement tient donc une place importante dans notre vie quotidienne. Celui-ci fait l'objet de nombreuses études. Parmi les premières analyses du mouvement cinématographique, nous connaissons celles de Eadweard Muybridge (né Edward James Muggeridge) ayant porté sur le mouvement du cheval puis celui des hommes ([110], [111]) ou encore les travaux de Etienne-Jules Marey [31]. Depuis, les techniques de capture du mouvement ont beaucoup évolué, que ce soit pour la biomécanique, l'animation de personnages de synthèse, le jeu vidéo, *etc.*

Dans le domaine de la réalisation de films animés ou de la production de jeux vidéo, la capture du mouvement prend une part de plus en plus importante. Les animateurs des jeux ou films sont particulièrement intéressés par le mouvement. Tout leur art consiste à créer un mouvement de personnage auquel le spectateur ou le joueur doit croire. Cela ne signifie par forcément que le mouvement soit réaliste, mais qu'il soit naturel et adapté au personnage. Cependant, l'animation de personnages humains requiert un réalisme particulier pour la plupart des applications. En effet, le personnage doit avoir une expression et un mouvement convaincant si le public doit s'identifier à celui-ci.

Beaucoup de séquences de films d'animations ont été copiées de scènes réelles filmées spécialement. La technique dite de rotoscopie (calquer le mouvement d'un personnage sur un mouvement filmé) employée à cet effet, développée par Max Fleischer, facilite la tâche des animateurs et permet d'obtenir des résultats très réalistes. C'est ce qui a été utilisé pour animer certaines scènes de Blanche Neige et les Sept Nains (des studios Disney), par exemple. Cette technique, de moins en moins utilisée, reste cependant un modèle pour les techniques modernes de capture du mouvement.

Dans ce chapitre, nous allons dans un premier temps aborder les systèmes de capture utilisés par les studios de productions. Trois grandes classes de systèmes sont utilisées : les systèmes à marqueurs magnétiques, les systèmes prosthétiques et les systèmes à marqueurs optiques. Après avoir décrit les deux premiers systèmes et donné les avantages et inconvénients de chacun d'eux, nous nous attarderons sur la description et les problématiques spécifiques aux systèmes optiques avec marqueurs. Nous verrons alors que la capture du mouvement sans marqueurs doit également résoudre ces problèmes ainsi que d'autres plus spécifiques à l'absence de marqueurs. Cette dernière catégorie de systèmes est l'objet de nombreux travaux en vision par ordinateur. Nous aborderons un état de l'art de ces systèmes en proposant différentes grilles de lecture des différents travaux.

2.1 Les systèmes prosthétiques et magnétiques

Que ce soit dans l'industrie du jeu, du film ou encore pour la médecine ou la biomécanique, les systèmes de capture du mouvement ont fait leur apparition pour aider à la production ou au diagnostique médical. Trois catégories de systèmes semblent prédominer actuellement dans ces domaines :

- Les systèmes électromécaniques prosthétiques. Un exosquelette muni de potentiomètres est attaché aux membres dont nous voulons analyser le mouvement. Ces systèmes paraissent imposants. Ils sont cependant encore très utilisés, car peu chers (de l'ordre de 25 000 €), fiables et assez simples à utiliser. Nous reviendrons dessus plus tard.
- Les systèmes à capteurs magnétiques. Des capteurs sensibles à un champ magnétique produit par une source sont posés sur les différentes parties du corps de l'acteur à suivre. Les données des capteurs sont traitées en temps réel ce qui permet de visualiser les résultats en temps réel. Contrairement aux systèmes prosthétiques, la gêne occasionnée par le système sur l'acteur est faible. En effet, seuls des câbles reliant les capteurs à une centrale portative entravent le mouvement de l'acteur.
- Les systèmes optiques. Ces systèmes utilisent des marqueurs réfléchissants ou spécifiques posés sur l'acteur. Des caméras sont utilisées pour détecter les marqueurs posés sur l'acteur. Il s'en suit une phase de reconstruction *3D* qui peut être temps réel. Ce type de système est très peu gênant pour l'acteur, puisque seuls des petits marqueurs sont posés sur l'acteur. Cependant, ce type de système impose des contraintes d'ordre vestimentaire ou encore pour l'environnement de capture (les marqueurs doivent être détectables le plus facilement possible).

Nous allons aborder rapidement les avantages et inconvénients de deux premiers systèmes. Nous nous attarderons sur la troisième catégorie dans la partie suivante. Pour plus de détails, le lecteur pourra se référer à [101]. Ce livre permet d'avoir une bonne introduction à la capture du mouvement telle qu'elle est abordée dans le milieu industriel.

Les systèmes prosthétiques Malgré son apparence archaïque comparée aux systèmes magnétiques ou optiques, ce type de système a des avantages qui font qu'il est encore utilisé dans certaines situations. Leur faible coût en fait un argument majeur pour les petites entreprises qui désirent un système de capture du mouvement. Il n'existe pas d'éléments perturbants pouvant empêcher le système de fonctionner alors que nous verrons que les systèmes magnétiques et optiques peuvent être perturbés lors d'une séance de capture du mouvement. Le champ d'action de l'acteur lors de la capture du mouvement est quasiment illimité, puisque l'exosquelette n'est relié à aucun système externe. Enfin, les mesures sont prises à l'aide de potentiomètres directement placés sur les articulations, ce qui simplifie les traitements pour la reconstruction du mouvement.

Cependant, il existe un inconvénient majeur, presque rédhibitoire, qui est l'encombrement du système et la gêne occasionnée pour l'acteur. Ce désavantage explique

FIG. 2.1: Système prosthétique de capture du mouvement (société Gypsy).

l'utilisation très restreinte pour la capture de mouvements rapides, amples ou sportifs (nécessitant une liberté de mouvement totale).

FIG. 2.2: Système de capture du mouvement FastTrack.

Les systèmes magnétiques Ce type de système est avantageux pour la capture du mouvement. En effet, les capteurs magnétiques sont conçus de sorte à ce que leur position et leur orientation soit mesurable. La phase de traitement des données et la génération du mouvement se trouve donc facilitées. De plus, la vitesse d'acquisition théorique est très rapide (de l'ordre de 100 Hz) et les données peuvent être traitées en temps réel ce qui permet d'obtenir un retour visuel pour l'acteur. Un mouvement peut donc être jugé exploitable ou non en temps réel ce qui est un avantage majeur. Le faible encombrement du système sur l'acteur et la facilité de sa mise en place sont aussi deux avantages. Cependant, ce système a des limites assez fortes. Les objets métalliques

peuvent entraîner des perturbations du champ magnétique produit par la source et donc des imprécisions dans la mesure du champ. Les environnements dans lesquels la capture est effectuée sont donc contraints. De plus, la source du champ magnétique à un champ de vue restreint, ce qui limite le champ d'action de l'acteur. Enfin, bien que rapide, la fréquence d'acquisition est limitée par le bruit de mesure, ce qui contraint généralement à filtrer les données pour arriver à une fréquence équivalente à du 15 Hz. Ce filtrage fait perdre toute possibilité d'utilisation pour la capture de mouvements sportifs rapides.

2.2 Les systèmes optiques à marqueurs

Plusieurs systèmes industriels sont actuellement proposés dans le commerce pour effectuer de la capture du mouvement avec marqueurs (VICON, BTS SMART-E, ISADORA, *etc.*). Nous avons travaillé à Rennes avec le système VICON.

Les problématiques rencontrées lors de l'utilisation (et probablement lors de la conception) de tels systèmes sont très proches de celles que nous rencontrerons pour effectuer la capture du mouvement multi-caméras sans marqueurs.

Le principe du système à marqueurs est de détecter des marqueurs posés sur l'acteur, d'effectuer dans un premier temps une reconstruction *3D* des marqueurs détectés et enfin de proposer une aide à l'estimation du mouvement. Le système dispose donc d'un nombre de caméras laissé à la guise de l'utilisateur (et de son porte-monnaie), de marqueurs passifs qui ne sont rien d'autre que de petites sphères réfléchissantes et d'une unité de traitement des données. Pour effectuer la détection des marqueurs, les caméras sont dotées de systèmes de diodes lumineuses éclairant le champ de vue de la caméra. Ces diodes ont une fréquence de clignotement identique à celle de la fréquence d'acquisition des caméras. Ce clignotement permet aux différentes caméras de ne détecter les marqueurs que s'ils sont éclairés par le faisceau lumineux du système de diodes attaché à la caméra. Les caméras filment donc une scène où des points très brillants apparaissent. Ces points sont détectés et leurs coordonnées sont transmises à une unité de calcul. Cette unité reçoit l'ensemble des positions des marqueurs détectés dans chacune des caméras et effectue une reconstruction *3D* de ces marqueurs. Cette étape nécessite quelques pré-requis : le système doit être calibré et la détection des marqueurs peu bruitée. Le calibrage du système se fait de manière automatique avant chaque séance d'acquisition. Un trièdre de marqueurs est posé au centre de la scène et le système se calibre en effectuant une reconstruction *3D* de cet objet. Le deuxième pré-requis est plutôt une contrainte, puisque d'une part l'environnement de capture ne doit pas comporter trop d'éléments réfléchissants et d'autre part, l'acteur doit être habillé de sorte à ce qu'il ne réfléchisse pas trop la lumière émise par les diodes et que les sphères réfléchissantes soient bien discernables dans les images. Typiquement, des vêtements clairs défavoriseraient une bonne mesure de la position des marqueurs.

L'ensemble de ces étapes pose des difficultés :

La détection des marqueurs : Les caméras utilisées pour visualiser les marqueurs sont généralement dotées de systèmes d'éclairage permettant de faciliter la

détection des marqueurs sur l'acteur. Cependant, la vitesse d'exécution des mouvements, la taille des marqueurs, les réflections multiples dans la scène posent de nombreuses difficultés pour rendre robuste cette détection.

La reconstruction des marqueurs : Une fois les marqueurs détectés dans chacune des images, il faut pouvoir les reconstruire en *3D*. Se pose alors un problème de mise en correspondance des marqueurs : un marqueur dans une image correspond il à un marqueur dans les autres images et si oui le quel ? Il y a donc un problème combinatoire à résoudre. Le problème est d'autant moins simple que les marqueurs ne sont pas discernables entre eux.

Le suivi des marqueurs : Une fois les marqueurs reconstruits, il faut pouvoir les suivre au cours du temps pour effectuer la reconstruction de trajectoire. Ce suivi pose de nombreux problèmes comme la mise en correspondance temporelle de marqueurs qui ne sont pas discernables mais aussi les problèmes d'occultation ou de disparitions de marqueurs.

La reconstruction des trajectoires angulaires : Enfin, cette dernière étape permet d'estimer les trajectoires de chacune des articulations. Pour cela, il faut pouvoir associer chaque marqueurs à une partie du corps. Il y a donc un problème d'assignation entre les marqueurs reconstruits et les parties du corps.

Pour aider à résoudre tous ces problèmes, certaines étapes sont nécessaires avant d'effectuer une capture du mouvement. Le système demande à l'acteur d'exécuter des mouvement spécifiques. Nous pensons qu'il s'agit d'un calibrage de l'acteur permettant d'aider au suivi des points *3D*. Dans un ordre prédéfini et indiqué par le système, l'acteur doit bouger les mains, les avant bras, les bras, le torse, les chevilles, les tibia, les cuisses, puis enfin la tête. Ces mouvements sont probablement nécessaires pour aider le système à positionner l'ensemble des marqueurs sur une structure de squelette aidant alors la phase de suivi. [114] propose une méthode pour estimer les paramètres articulaires d'un squelette à partir des données issues d'un système avec marqueurs (magnétiques dans ce cas). [69] souligne la difficulté à utiliser les marqueurs pour estimer les paramètres du mouvement et propose d'intégrer l'utilisation d'un squelette pour aider au suivi et à l'estimation.

Lors de séances de capture de mouvements simples, les post-traitements peuvent être très rapides (de l'ordre de 10 à 15 minutes pour une acquisition de 2 à 3 minutes). Cependant, lors de séances d'acquisitions de mouvements complexes, les post-traitements peuvent s'avérer très long et demander à des utilisateurs experts de retoucher les séquences de reconstruction du mouvement pendant des jours.

L'avantage majeur de ces systèmes est l'absence de câbles ou de structure métallique pouvant gêner le mouvement de l'acteur. Cependant, les marqueurs ou la tenue vestimentaire peuvent encore gêner certains mouvements comme ceux de sportifs. La vitesse d'acquisition est suffisamment rapide pour effectuer la capture de mouvements sportifs. La surface utile pour l'acquisition est aussi plus grande que pour les systèmes magnétiques, puisqu'elle dépend du nombre de caméras utilisées. Parmi les inconvénients de ces systèmes, nous pouvons citer le temps de calcul nécessaire pour visualiser une séquence d'acquisition. En effet, le nuage de marqueurs est reconstruit en temps réel,

mais le squelette n'est pas observable. Il est donc difficile de se rendre compte de l'utilisabilité d'une séquence de mouvement pendant l'acquisition. De plus, alors que les systèmes magnétiques permettent d'obtenir la position et l'orientation des différentes articulations, les systèmes optiques sont limités à la position. Pour pouvoir obtenir l'orientation des différentes articulations, le nombre de marqueurs doit être augmenté (par exemple deux ou trois marqueurs pour le poignet), ce qui augmente aussi les temps de calcul et de synthèse du mouvement articulé. Enfin, le coût de ces systèmes est élevé : celui-ci varie en fonction du nombre de caméras acquis, mais peut s'élever à 400 000 € pour un système doté de 12 caméras.

Ces systèmes utilisés dans l'industrie du jeu permettent de faciliter la production de jeux vidéo. Ce sont des systèmes qui s'avèrent coûteux et qui nécessitent donc d'être rentabilisés. De petites sociétés ne peuvent acheter de tels systèmes et si besoin est loue un temps d'utilisation à d'autres sociétés ayant ce type de système. Cette location est motivée par le faible coût d'une part et la possibilité d'avoir des experts pour utiliser le système. Cependant, ce mode d'utilisation n'est pas optimal, puisque la marge d'erreur lors d'une séance de capture est très faible : refaire une scène nécessite de relouer le système.

L'évolution de la puissance de calcul des ordinateurs, ainsi que l'évolution du matériel d'acquisition vidéo aujourd'hui accessible pour des petites et moyennes structures a poussé la recherche à élaborer des systèmes de capture de mouvement sans marqueurs. Outre la motivation financière, ces systèmes s'affranchisse alors de toute contrainte vestimentaire pour l'acteur, le sportif ou encore le patient (pour l'étude de pathologies moteurs par exemple). Les nouveaux défis sont donc d'effectuer de la capture du mouvement non invasive et où les contraintes sur l'environnement de capture sont très faibles. Nous allons maintenant aborder la problématique de la capture du mouvement sans marqueurs.

Synthèse : Nous avons décrit un système à marqueur type. Nous avons vu que différentes étapes étaient nécessaires pour effectuer une séance de capture du mouvement :

- La mise en place du système et son calibrage,
- Le positionnement des marqueurs sur le corps de l'acteur et le calibrage de ces marqueurs par le système,
- La détection automatique des marqueurs dans les images et la reconstruction *3D* de ces marqueurs,
- L'estimation du mouvement des différentes parties du corps,
- Le post-traitement des données pour améliorer la qualité de l'estimation.

Ce protocole opératoire va constituer le fil directeur de notre état de l'art.

2.3 Systèmes optiques sans marqueurs

Dans le paragraphe précédent, nous avons abordé une description extensive du système de capture utilisant des marqueurs. Nous allons maintenant nous intéresser

(a) (b) (c)

FIG. 2.3: Système de capture du mouvement VICON. (a) illustre la capture du mouvement du corps humain complet (source : http ://en.wikipedia.org/wiki/Motion_capture). (b) illustre la capture du mouvement expressif du visage (source : http ://www.spgv.com/columns/motioncapture.html). (c) illustre de la capture du mouvement animale (source :http ://home.blarg.net/w̃ayule/graphics/motioncapture.jpg)

à la capture sans marqueurs. Nous allons montrer dans un premier temps que les problématiques de la capture sans marqueurs sont similaires à celles posées par les systèmes avec marqueurs. Après avoir identifié les problématiques, nous proposerons différentes grilles de lecture des travaux précédents. Nous montrerons que les approches multi-caméras ne sont pas plus simples que les approches monoculaires puisque les contraintes et les objectifs sont différents. L'essentiel du propos de cet état de l'art se référera aux approches multi-caméras. Nous aborderons la modélisation du corps humain, la mise en correspondance du modèle avec les données ainsi que l'estimation des paramètres du mouvement de l'acteur. Nous finirons cet état de l'art en mettant en avant des critères que nous pourrions prendre compte pour évaluer les résultats de la capture du mouvement sans marqueurs.

2.3.1 Identification des problèmes

Tout comme pour la capture du mouvement à l'aide de systèmes avec marqueurs, différentes étapes sont nécessaires pour effectuer de la capture sans marqueurs, et pour chacune de ces étapes plusieurs difficultés sont à résoudre. Nous allons maintenant montrer le parallèle entre les deux approches.

- Le nombre de caméras utilisées. Deux grandes classes d'approches peuvent être distinguées : celles utilisant plusieurs caméras et celles n'en utilisant qu'une. Les problématiques posées par chacune des deux approches s'avèrent différentes. Nous traiterons dans cet état de l'art plus particulièrement des approches utilisant plusieurs caméras. Cependant, nous consacrerons un paragraphe sur les approches mono-caméra dans la section 2.3.2. Lors de l'utilisation du système à marqueurs, il est nécessaire d'utiliser plusieurs caméras pour effectuer la reconstruction *3D*.
- Lors de l'utilisation de plusieurs caméras, le premier problème qui se pose est le

positionnement des caméras. Il s'agit d'avoir le meilleur recouvrement possible de la scène à capturer ainsi qu'une résolution suffisante de l'acteur. Une fois les caméras positionnées, nous devons les calibrer.
- Nous avons vu que le système à marqueurs demande dans un premier temps à l'acteur d'effectuer certains gestes pour pouvoir calibrer la reconstruction *3D* des trajectoires angulaires. Dans un système sans marqueurs, la question se pose de façon encore plus aiguë : comment modéliser l'acteur pour effectuer le suivi. Ce choix dépend des méthodes de suivi du mouvement employées. Le choix de la représentation porte aussi bien sur la structure cinématique du modèle (le squelette de l'acteur) que sur le modèle d'apparence. Une fois la modélisation choisie, il s'agit de faire en sorte que le modèle d'apparence corresponde au mieux à l'acteur suivi. Il s'agit d'une phase de *capture de l'acteur*.
- Pour effectuer le suivi du mouvement, un système à marqueurs détecte dans les images les marqueurs posés sur l'acteur par simple détection de maxima d'intensité. En l'absence de marqueur, il faut déterminer quelles observations seront extraites des images. Ces observations permettront de comparer le modèle de l'acteur aux images. Ces informations peuvent être de différentes sortes : les silhouettes, les contours, la couleur, des marqueurs spécifiques, des points d'intérêts détectés mais aussi des informations de plus haut niveau comme la texture, le flot optique, *etc*.
- Une fois les marqueurs détectés, les systèmes à marqueurs effectuent une triangulation pour les reconstruire en *3D*. De manière générale, la question posée dans cette étape est la mise en correspondance des données extraites des images avec le modèle de l'acteur. Nous verrons différents types d'approches.
- Enfin, la dernière étape consiste à effectuer l'estimation des trajectoires angulaires. Pour les systèmes avec marqueurs, cela nécessite un calibrage des positions des marqueurs dans les référentiels locaux des segments ; Puis la résolution d'un problème de cinématique inverse dont la solution peut être obtenue en résolvant un système de moindre carré non linéaire. Dans le cadre de la capture sans marqueurs, nous pouvons adopter une approche similaire. Cependant, le problème est d'autant plus compliqué que les données sont bruitées, incomplètes et imprécises.

Après avoir rapidement abordé la capture du mouvement utilisant une seule caméra, nous nous attarderons sur les problématiques de la capture du mouvement multi-caméras. Nous aborderons dans un premier temps la problématique de la modélisation de l'acteur et du dimensionnement du modèle. Puis nous verrons que pour effectuer la mise en correspondance, nous pouvons distinguer trois grandes classes d'approches. Enfin, nous aborderons les différentes approches proposées pour estimer les paramètres de pose de l'acteur.

2.3.2 Mono vs. Multi Caméras

De manière générale, la littérature sur la capture du mouvement distingue deux approches : les systèmes mono-caméra et les systèmes multi-caméras. Dans un premier temps nous aborderons un rapide état de l'art sur les approches mono-caméra. La suite

de ce chapitre sera dédiée aux problématiques essentiellement liées aux approches multi-caméras.

Systèmes mono-caméra Les approches mono-caméra pour la capture du mouvement trouvent leur motivation dans la volonté de pouvoir effectuer l'estimation du mouvement à partir de séquences vidéo provenant de scènes de films, de caméras de vidéo-surveillance, *etc*. L'utilisation d'une unique caméra pour effectuer le suivi du mouvement est un véritable défi. En effet, le mouvement observé à partir d'un unique point de vue peut être ambiguë. D'une part parce que l'estimation de la profondeur n'est pas possible et d'autre part parce que les occultations ne peuvent que difficilement être prises en compte. La problématique générale posée par ce type d'approche est la mise en correspondance de données *3D* du monde avec une représentation planaire de celui-ci. Pour pouvoir effectuer le suivi du mouvement avec un tel système, il est donc nécessaire d'introduire des contraintes dans l'estimation du mouvement. Les premières approches se sont concentrées sur l'étude de mouvements cycliques ou parallèles au plan image de la caméra. [75] propose la première approche pour effectuer le suivi du mouvement. [149] propose par exemple d'utiliser un filtrage de Kalman ([150]) avec un modèle de contraintes cinématiques pour effectuer la capture du mouvement de la marche. [79] propose d'utiliser une représentation du corps humain sous forme de *patch* planaire. La contrainte entre ces différents patchs est exprimée sous la forme d'une contrainte de type loi de Hook entre les sommets des différents patchs. Le mouvement dans l'image est estimé en utilisant le flot optique. [107] (puis dans [123]) propose d'utiliser un modèle *2D* dit prismatique caractérisant un modèle *3D*. Ce modèle *2D* permet de s'affranchir de singularités pouvant apparaître en utilisant directement le modèle *3D*. D'autres approches existent comme celle proposée dans [125] (*pictorial structures*).

Pour pouvoir traiter des mouvements complexes, une tendance actuelle consiste à introduire des connaissances a priori sur le mouvement. [2] propose d'utiliser les SVM (*Support Vector Machine* [147]) pour effectuer le suivi du mouvement à partir de silhouettes. Certains travaux proposent d'intégrer des positions et des images clefs dans les séquences ([41], [96]). Enfin, peu de travaux proposent des méthodes de suivi de mouvements complexes sans apprentissage et sans intervention de l'utilisateur. [129] propose d'utiliser des méthodes de *bundle adjustment* pour effectuer le suivi *3D*. Nous pouvons aussi citer [136] qui propose une approche utilisant un modèle *3D* (composée de super-quadriques) pour effectuer le suivi de mouvements non contraint. La méthode est dérivée du filtrage particulaire. Il s'agit de générer des hypothèses de pose, de les comparer aux images en utilisant une distance de chanfrein, et d'affiner par minimisation directe les poses les plus proches de celles observées (dans les images).

Systèmes multi-caméras Pour éliminer les ambiguïtés présentes avec le suivi mono-caméra, des systèmes utilisant plusieurs caméras sont apparus. Les motivations deviennent alors différentes de celles des approches mono-caméra. Les scènes filmées sont dédiées à la capture du mouvement. Les objectifs sont alors la synthèse de mouvement

pour l'animation ou encore l'étude précise du mouvement (que ce soit pour le sport ou pour la médecine, par exemple).

L'accroissement du nombre de caméras a été permis d'une part par la baisse des coûts du matériel (caméras, unités de stockage, *etc.*) ainsi que par l'augmentation de la puissance de calcul des ordinateurs. La notion de suivi multi-caméras est apparue avec l'utilisation de paires stéréo ([9], [36], [119], [11]), des grilles de caméras [30] ou encore plusieurs caméras calibrées ([115], [52], [82], [71], [38], [34], [45], [104], [141], [103], [25], [128], [20]). Pendant ma thèse, de nombreux travaux ont aussi été publiés sur la capture multi-caméras ([138], [113], [83], [102], [17], [84], [10], etc.).

L'utilisation de plusieurs caméras permet de désambiguïser l'estimation du mouvement. En effet, les problèmes d'estimation de profondeur, de modélisation d'apparence, d'occultations sont mieux conditionnés que dans les approches mono-caméra. Cependant les défis relevés sont d'un autre ordre et ne simplifient pas moins le problème. En effet, la précision, les temps d'exécution, la robustesse sont autant de critères nécessaires pour que de tels systèmes puissent être utilisés dans le monde de l'animation ou de l'étude du mouvement humain.

2.3.3 Placement et calibrage des caméras

Peu de travaux abordent la problématique du placement des caméras. Nous pouvons citer [82] qui propose de placer trois caméras dans des positions mutuellement orthogonales. Ce positionnement permet un sélection intelligente du point de vue utilisé pour effectuer l'initialisation du modèle ainsi que la capture du mouvement. Le positionnement des caméras est important lors de la mise en place d'un système de capture du mouvement. D'une part, il doit dépendre du type de mouvement effectué lors de la séance de capture. D'autre part ce placement doit être optimal pour l'acquisition des images. En effet, de la position des caméras dépend la qualité des images obtenues (spots lumineux présents dans les images, contre jour, mauvais éclairage, faible luminosité, vue sur une fenêtre extérieure, *etc.*) et donc la précision des résultats. L'autre point important qui est le calibrage des caméras. Les techniques pour effectuer le calibrage des caméras sont nombreuses. Certaines approches proposent l'utilisation de mires (comme celle d'OpenCV), d'objets comme des trièdres (vicon), des bâtons pré-calibrés (ce que nous utilisons sur la plate-forme d'acquisition multi-caméras à l'inria [1]). Certains travaux tentent maintenant de s'affranchir de calibration ([142]) ou tout du moins proposent des techniques d'auto-calibration en utilisant par exemple des silhouettes extraites des images ([16]). Ce calibrage est bien évidement primordial pour la précision des résultats de la capture du mouvement, et plus précisément pour l'étape de la mise en correspondance des observations avec le modèle.

[1] Grimage : http ://www.inrialpes.fr/sed/grimage/

2.3.4 Modélisation de l'acteur

Pour effectuer le suivi du mouvement humain, il est nécessaire de choisir une représentation paramétrée de l'acteur. De manière générale, les approches multi-caméras utilisent des représentations tri-dimensionnelles des acteurs. Ces représentations comprennent deux aspects : d'une part la représentation volumétrique (ce que nous aborderons au paragraphe 2.3.4.2) et d'autre part la représentation cinématique du squelette (ce que nous abordons maintenant).

2.3.4.1 La représentation cinématique du squelette

La chaîne cinématique permet de décrire les contraintes du mouvement de l'ensemble des parties du corps. Elle correspond au squelette de l'homme. La représentation de ce squelette peut être plus ou moins complète selon le degré de précision requis pour l'application. L'ensemble des articulations du squelette peut être représenté avec différents formalismes, sur lesquels nous allons nous attarder maintenant.

Dans la littérature, nous pouvons distinguer deux grandes approches pour la modélisation des articulations : une représentation proche de la représentation robotique et une représentation que nous pouvons qualifier de représentation en coordonnées cartésiennes.

Représentation standard La représentation cinématique la plus couramment utilisée pour effectuer le suivi du mouvement est une représentation où les différentes parties du corps sont organisées de manière hiérarchique, le lien entre les différents niveaux se faisant par des articulations (typiquement des liaisons de type rotule, cardan, *etc.*). Les différences entre les divers travaux se situent alors au niveau de la représentation des articulations. Chaque articulation du squelette peut comporter 1, 2 ou 3 degrés de liberté (d.d.l.) en rotation. Il s'agit donc de modéliser ces d.d.l. Trois représentations sont possibles :

Euler : Il s'agit probablement de la représentation la plus intuitive des rotations pour la modélisation d'une chaîne cinématique. Chaque articulation est alors modélisée à l'aide de 3 valeurs dont 1, 2 ou les 3 peuvent varier, ce nombre dépendant du nombre de d.d.l. Cette représentation est très utilisée car simple à mettre en place ([53], [34], [136], [70], etc). Cette modélisation souffre cependant de singularités (le *Gimbal Lock*) comme nous le verrons au chapitre 3 dans le paragraphe 3.2.3.

Quaternions : Pour palier aux problèmes de singularités rencontrés avec l'utilisation des angles d'Euler, des travaux proposent d'utiliser les quaternions. Leur représentation est très compacte et leur utilisation ne nécessite pas de calculs trigonométriques lourds pour exprimer une chaîne cinématique. [82] les utilise pour paramétrer les articulations du modèle *3D*. [70] utilise les quaternions pour modéliser les articulations ayant trois d.d.l. (mais utilise les angles d'Euler sinon).

Axe et angle : Cette représentation est très utilisée en robotique [109]. Elle s'appuie sur le fait que toute rotation peut être exprimée à l'aide d'un axe de rotation

et d'un angle associé. Ce formalisme a été introduit en capture du mouvement dans [19] (puis repris dans [20]) pour représenter de manière compacte la chaîne cinématique du corps humain. D'autre travaux comme [35] ou encore [103] utilisent aussi cette représentation, que nous adoptons également.

Représentation en coordonnées cartésiennes Une deuxième manière de représenter la chaîne cinématique est de considérer les différentes parties du corps indépendantes les unes des autres et de contraindre le positionnement de chacune en fonction de contraintes cinématiques. Ces contraintes sont généralement exprimées sous la forme de contraintes probabilistes. [48] propose d'utiliser les *pictorial structures* pour effectuer le suivi du mouvement dans une séquence vidéo. Chacun des membres est représenté par une région planaire rectangulaire (*patch*) modélisant l'apparence. Les *patchs* sont reliés entre eux par des contraintes de type ressort. [120] propose d'utiliser des modèles graphiques pour contraindre la pose des différentes parties du corps détectées dans les images. Alors que les deux approches précédentes sont des approches mono-caméra, [134] propose une extension au cas *3D*.

[36] (étendu dans [37]) propose de suivre chaque partie du corps de manière indépendante. La pose de chacune des parties du corps est estimée sans contraintes dans un premier temps à partir d'une reconstruction stéréo. Puis les paramètres estimés sont projetés dans l'espace des configurations possibles (cet espace étant préalablement appris) et modifiés pour satisfaire les contraintes cinématiques.

2.3.4.2 Représentation volumétrique

Il existe autant de modèles *3D* d'acteurs que de travaux publiés sur le suivi du mouvement. Ces modèles varient aussi bien en terme de formes de primitives qu'en terme de nombre de primitives utilisées pour modéliser le corps humain. Le nombre de primitives du modèle est généralement lié au nombre d'éléments que comprend la chaîne articulaire. Cependant, certains auteurs multiplient le nombre de primitives dans la représentation de leur modèle *3D* sans augmenter le nombre de d.d.l. de la chaîne articulaire. L'objectif est alors d'affiner la représentation du modèle pour améliorer le suivi du mouvement.

Les primitives Les types de primitives utilisées pour modéliser le corps humain dépendent de la précision et de la méthode de suivi mise en place. De manière générale, nous pouvons distinguer les modèles rigides des modèles déformables. Nous donnerons les principaux modèles formés de primitives rigides pour effectuer la capture du mouvement. Puis nous aborderons rapidement le cas des modèles déformables.

Le modèle le plus simple est celui composé de bâtons (*stick model*). Il ne sert généralement que de support pour la chaîne articulaire et est généralement associé à des méthodes de reconstruction de modèle *3D* à partir des observations. Dans [102], les auteurs proposent une méthode utilisant ce type de modèle, où la pose de l'acteur est

estimée à partir d'une reconstruction d'enveloppes visuelles (c.f. figure 2.4-d). D'autres travaux utilisent le modèle de type bâton pour apporter une information sur la pose de l'acteur. [17] propose une méthode qui effectue simultanément une segmentation d'image et l'estimation de la pose d'un acteur. Le modèle permet alors d'apporter une connaissance a priori sur l'acteur dans les images. L'algorithme proposé utilise les *Graph Cuts* dynamiques ([90]) pour segmenter et estimer la pose de l'acteur.

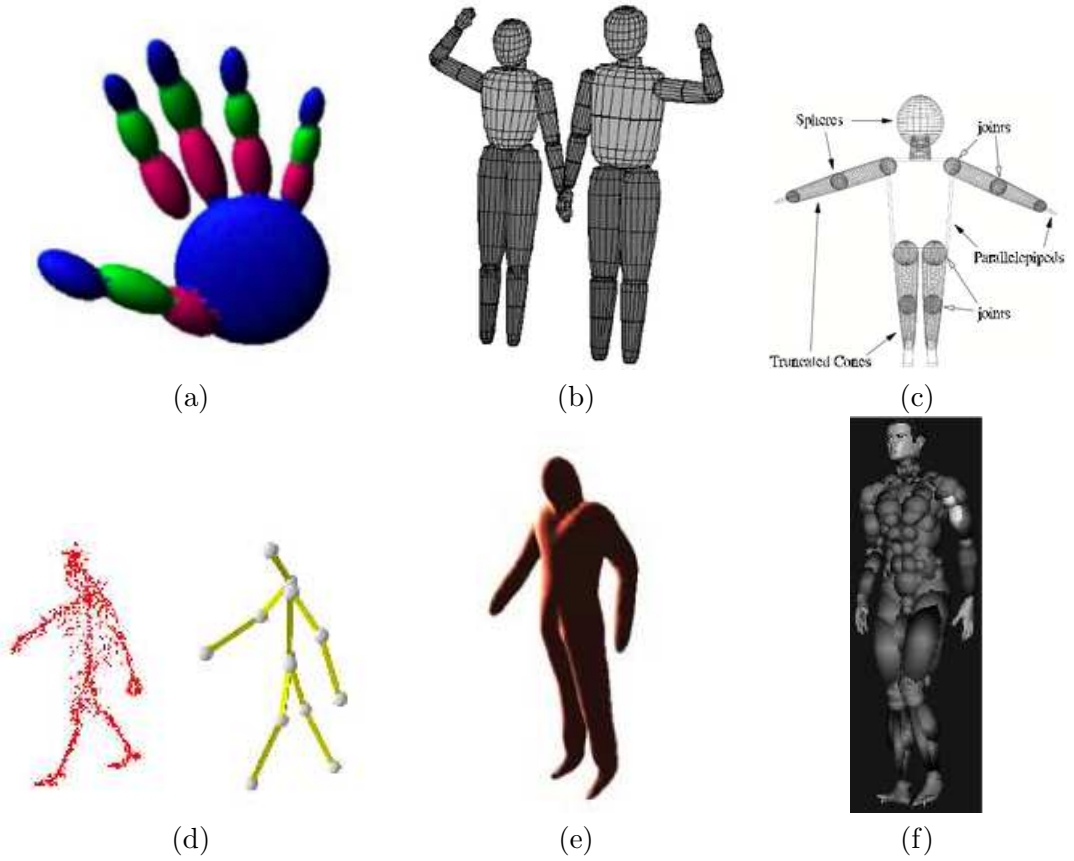
Les quadriques sont plus largement utilisées pour modéliser les différentes parties du corps. Par exemple [137] décrit un modèle (de la main) composé de 39 quadriques tronquées. [34] propose un modèle composé de cônes et de sphères pour modéliser la main ou le corps humain en entier (c.f. figure 2.4-c). [104], [38], [132], [85], [10] proposent d'utiliser un modèle composé de troncs de cônes à base circulaire ou elliptique. Enfin, les ellipsoïdes bien réparties sur le modèle modélisent de manière correcte la morphologie humaine ([103]).

Une représentation plus précise mais plus complexe à utiliser est celle des superquadriques. [77] propose une description extensives de ces primitives géométriques. [52] et [138] proposent d'utiliser ces primitives pour modéliser le corps humain en entier (c.f. figure 2.4-b).

Certains travaux proposent d'utiliser des modèles déformables. Par exemple [82] propose d'utiliser un modèle déformable pour capturer les dimensions de l'acteur. Ce modèle est ensuite utilisé (et localement modifié) pour effectuer le suivi du mouvement. [113], tout comme [39] pour le suivi de la main, utilisent les surfaces implicites construites à partir d'ellipsoïdes (c.f. figure 2.4-(a) et 2.4-e). La surface utilisée lors du suivi de mouvement est donc déformable, mais utilise des ellipsoïdes qui sont eux des primitives rigides. Cette modélisation permet de rendre compte de certaines déformations de la peau tout en conservant la possibilité de traiter le modèle à l'aide d'une chaîne cinématique simple.

Enfin, [119] propose une représentation hiérarchique du modèle. Les auteurs utilisent des *méta-balls* ellipsoïdaux (c.f. figure 2.4-f) ainsi que des ellipsoïdes pour modéliser les parties du corps. La représentation du modèle comporte deux niveaux de détails : le premier permet de modéliser le corps humain de manière grossière et d'effectuer le suivi. Le second niveau permet de modéliser plus finement le corps pour pouvoir créer un modèle réaliste. Le modèle est construit en utilisant la technique proposée dans [140].

Le nombre de primitives La complexité des modèles *3D* dépend non seulement de la complexité de la chaîne cinématique mais aussi de la finesse avec laquelle le corps humain est modélisé. A nombre d'éléments de la chaîne cinématique constant, nous pouvons augmenter le nombre de primitives géométriques pour modéliser le corps humain. Nous avons vu dans le paragraphe précédent que le modèle développé dans [140] comprenait plusieurs niveaux de détail. Le modèle de la main proposé dans [137] comporte des primitives rigidement liées les unes aux autres ce qui permet par exemple de modéliser la paume de la main correctement.



(a) (b) (c)

(d) (e) (f)

FIG. 2.4: Quelques exemples de modèles *3D* utilisés dans la littérature. (a) est construit à partir d'ellipsoïdes ([39]), (b) à partir de super-quadriques ([52]), (c) à partir de tronc de cônes et de sphères ([34]). (d) est un modèle de type bâton ([102]), (e) est un modèle de type surface implicite construite à partir d'ellipsoïdes ([113]). Enfin (f) est le modèle proposé dans [119], construit avec des méta-balles.

2.3.5 Mise en correspondance

Dans la partie précédente, nous avons abordé la modélisation cinématique et volumétrique du corps humain. Nous allons maintenant aborder la problématique de la mise en correspondance des données avec le modèle. Nous pouvons distinguer deux grandes classes de méthodes. La première utilise directement des données dans les images et compare donc le modèle aux images. La seconde utilise une représentation *3D* des données extraites des images, le modèle est alors mis en correspondance avec cette reconstruction *3D*. Récemment, certains travaux ont proposé d'allier ces deux types d'approche.

Approche *2D* → *2D* Cette approche est surtout rencontrée dans le cadre de la détection de personnes ou les applications de vidéo-surveillance. Il s'agit d'une approche de type mono-caméra, ce que nous avons abordé au paragraphe 2.3.2. Le nombre de caméras alors utilisées n'influe pas sur la méthode de détection ou de suivi. Chacune des caméras est utilisée de manière indépendante.

Approche *2D* → *3D* Un modèle *3D* est utilisé pour effectuer le suivi du mouvement à l'aide de une ou plusieurs caméras. Le modèle *3D* est projeté sur les images. Il s'agit donc d'estimer l'erreur entre la projection et les observations. Ce type d'approche à l'avantage de pouvoir utiliser les informations directement dans les images comme les contours, la couleur, mais aussi la texture ou encore le flot optique.

[52] propose d'utiliser les contours détectés dans les images à l'aide d'un filtre de Canny [22]. Le modèle *3D* (composé de super-quadriques) est projeté dans les différentes images. Il s'agit donc de comparer les contours projetés du modèle avec les contours extraits dans les images. Afin de n'utiliser que les contours utiles extraits des images, ils sont filtrés : d'une part les contours trop éloignés de la prédiction donnée par le modèle (projection à l'instant t du modèle dont la pose a été estimée à l'instant $t-1$) ne sont pas pris en compte, et d'autre part les contours statiques au cours de la séquence sont éliminés. L'erreur entre la projection du modèle et les contours extrait de l'image est calculée en utilisant la distance de Hausdorff ([76]), avec comme métrique associée, la transformée en distance (distance de chanfrein dirigée). Nous utiliserons une méthode très similaire à celle-ci pour calculer l'erreur entre la projection de notre modèle et les contours détectés dans les images.

[71] propose d'utiliser quatre caméras mutuellement orthogonales pour effectuer le suivi du mouvement. Le mouvement d'un point de l'espace est estimé en calculant le flot optique dans chacune des images. Une relation linéaire entre le déplacement *3D* de points dans l'espace et le déplacement de ces points observés dans l'image permet de déterminer le mouvement du modèle.

[38] puis [33] proposent d'utiliser les silhouettes et les contours détectés dans les images pour effectuer l'estimation du mouvement. Chacune de ces images est lissée en utilisant un filtre gaussien. Cette carte lissée joue le rôle d'une carte de proximité (0 si

on est loin du contour et 1 dessus). Le modèle est projeté dans l'image et la distance entre la projection et les observations est calculée en sommant les valeurs lues le long des contours projetés (méthode similaire à [52]).

[45] propose aussi d'utiliser les contours dans les images. Cependant, les contours images ne sont pas détectés de manière standard. Le modèle *3D* est projeté dans les images. En chacun des points du contour projeté, la normale (au contour) est calculée. Alors, les contours images ne sont détectés que le long de ces normales. L'erreur entre le modèle projeté et l'observation est alors calculée sous la forme d'une distribution de probabilité dépendant de la distance d'un point du contour modèle au point de contour détecté sur la normale associée. Les auteurs obtiennent ainsi la probabilité d'observer une image étant donné la projection du modèle.

[34] propose d'utiliser la silhouette extraite des images pour effectuer le suivi du mouvement. L'approche proposée utilise une formulation dynamique du problème de la mise en correspondance : des forces d'attraction sont calculées entre les contours projetés (du modèle) et les contours observés. L'objectif est alors de faire en sorte que le bilan des forces soit nul après estimation de la pose.

[20] reprend les travaux proposés dans [19]. Les auteurs proposent l'extension d'une approche monoculaire pour effectuer la capture du mouvement avec plusieurs caméras. Pour effectuer le suivi, les auteurs utilisent l'estimation robuste du mouvement dans les images ([14]). Or le mouvement de l'acteur observé dans l'image est fonction de la variation des paramètres de pose de celui-ci. Une relation explicite entre le mouvement observé et la variation des paramètres de pose est donc proposée.

[132] propose d'utiliser les travaux introduits dans [131] pour effectuer le suivi du mouvement. La pose de l'acteur est estimée en utilisant un fitrage particulaire généralisé. Le modèle *3D* est mis en correspondance avec les images en utilisant un modèle d'apparence permettant de distinguer l'acteur de la scène ainsi que les différents membres de l'acteur. C'est cette modélisation qui est introduite dans [131].

[17] propose d'effectuer simultanément la segmentation des images et l'estimation de la pose de l'acteur. L'utilisation de chaînes de Markov pour évaluer la fonction de coût à minimiser permet d'introduire plusieurs mesures comme les contours, le fond de l'image et les silhouettes. [10] propose d'utiliser un modèle d'apparence pour effectuer le suivi du mouvement humain.

Un des problèmes posé par ce type d'approche est la gestion des occultations lors de l'estimation des paramètres de pose. En effet, la projection du modèle dans les images doit prendre en compte la visibilité ou non de certaines parties du corps. Pour remédier à ce problème, [82] propose d'utiliser un système de trois caméras en position mutuellement orthogonale avec une sélection du point de vue le plus adéquat pour estimer la pose. Cependant, avec l'évolution des capacités de calcul des cartes graphiques, nous pouvons maintenant utiliser ces dernières pour gérer de manière rapide les occultations.

Approche *3D* → *3D* Il s'agit de mettre en correspondance le modèle *3D* avec des données *3D*. Les données *3D* peuvent être obtenues par l'utilisation de paires stéréo.

Par exemple, [36] propose d'utiliser une paire stéréo pour effectuer le suivi des bras pour un système d'interaction homme-machine. La carte de profondeur est mise en correspondance avec un modèle *3D* en utilisant une méthode de type ICP (Iterative Closest Point [12]). [119] propose d'utiliser les données stéréo pour dimensionner un modèle *3D* et effectuer le suivi du mouvement. La distance entre les données et les observations est calculée de manière explicite entre les métaballs et les données stéréo. [39] propose d'utiliser une paire stéréo pour effectuer le suivi de la main. L'erreur est calculée en prenant en compte une distance point (du modèle) à point (obtenu par reconstruction) et une distance point (obtenu par reconstruction) à surface (du modèle). Cette double contribution permet de rendre robuste le suivi de la main.

Les méthodes de *shape from silhouettes* permettent d'effectuer le suivi du mouvement humain. [28] effectue une reconstruction voxéllique ([139]) puis estime les paramètres d'un modèle *3D* qui approche au mieux le volume reconstruit. [26] et [27] proposent une étude approfondie de la capture du mouvement en utilisant l'approche développée dans [28]. [103] utilise aussi les voxels avec un modèle *3D* complexe du corps humain. Parallèlement à notre travail, [102] propose d'utiliser les enveloppes visuelles ainsi qu'un modèle de type *stick figure* (bâton) pour effectuer la capture du mouvement. Chacun des segments du modèle est associé aux sommets de l'enveloppe visuelle permettant ainsi de calculer une distance représentative de la qualité de l'estimation des paramètres de pose. [83] utilise les enveloppes visuelles pour effectuer le suivi du mouvement. Pour calculer l'erreur entre le modèle et la reconstruction, les auteurs proposent d'utiliser une technique dite d'échantillonnage aléatoire. Un modèle *3D* constitué de points est mis en correspondance avec la reconstruction *3D* faite à partir des silhouettes. Les points du modèle *3D* sont sélectionnés de manière aléatoire afin d'aider la minimisation à sortir de minima locaux. Pour améliorer le suivi les auteurs proposent d'utiliser aussi de la couleur. Cette approche permet un assignement des parties du corps plus efficace dans des cas difficiles (proximité de deux parties du corps). Enfin, [113] propose d'utiliser non pas les enveloppes visuelles mais des points *3D* et les normales à l'enveloppe en ces points pour pouvoir utiliser une approche comparable à celle proposée dans [39].

Approches mixtes [104] propose d'estimer le mouvement de plusieurs personnes simultanément. Les auteurs proposent d'utiliser les enveloppes visuelles calculées à partir des silhouettes ainsi que les contours extraits des images. [141] et [24] proposent une approche en deux étapes. Lors de la première étape, les mains, les pieds et la tête sont suivis dans chacune des images et la position de chacun de ces membres est reconstruite en *3D*. Un squelette peu précis est alors mis en correspondance avec ces reconstructions pour estimer une position approximative de l'ensemble des membres. Dans une deuxième étape, l'enveloppe visuelle permet d'estimer les paramètres d'un squelette dérivé du premier et ayant un plus grand nombre d'articulations. Cette seconde étape permet d'estimer la pose correcte de l'acteur. [84] propose d'utiliser l'approche exposée dans [83] en ajoutant une contribution des contours détectés dans les images. La prise en compte de ces contours permet de désambiguïser l'estimation de la pose dans le cas où les silhouettes ne permettent pas d'avoir suffisamment d'information sur la position et l'orientation de chacun des membres.

2.3.6 Estimation du mouvement

Nous avons présenté différentes possibilités pour modéliser l'acteur et nous avons abordé différentes méthodes de mise en correspondance entre le modèle et les observations. Nous allons maintenant aborder les aspects numériques du suivi du mouvement : l'estimation des paramètres de pose d'un acteur ou d'un objet articulé au cours d'une séquence vidéo. L'estimation robuste et précise des paramètres de pose est le point clef de la capture du mouvement. Plusieurs approches ont été proposées. Nous pouvons en distinguer trois grands types.

Approches *Generate and Test* : Ce type d'approche est aussi appelée « méthode générative ». Il s'agit de générer des jeux de paramètres de pose du modèle et de vérifier que la pose ainsi générée corresponde aux observations. [52] utilise cette approche. La fonction de coût à minimiser dépend de la projection du modèle dans les images et donc des paramètres de pose du modèle. L'auteur propose donc d'effectuer une recherche des paramètres de paramètres de pose en incrémentant chacun des paramètres. A chaque incrément, la pose est recalculée et la fonction de coût ré-estimée. Si la fonction a diminuée, le jeu de paramètre est retenu. Cependant, cette recherche pas à pas de la solution est extrêmement coûteuse. L'auteur propose donc d'effectuer une recherche des paramètres dans un espace contraint. De plus, les différents paramètres sont recherchés avec un ordre hiérarchique (d'abord le pelvis, puis le torse, puis les bras, les jambes, *etc.*). Peu d'auteurs ont utilisé (à notre connaissance) ce type d'approche du fait du coût en temps de calcul.

Approches bayésiennes : Plusieurs travaux proposent d'utiliser des méthodes bayésiennes qui cherchent le maximum de la fonction de vraisemblance $P(\boldsymbol{I}|\boldsymbol{\Phi})$ (ce qui est en pratique compliqué à calculer) ou encore le maximum a posteriori $P(\boldsymbol{\Phi}|\boldsymbol{I})$ où $\boldsymbol{I}$ est l'ensemble des images à l'instant t et $\boldsymbol{\Phi}$ le vecteur des paramètres de pose de l'acteur. [38] propose d'utiliser un filtrage particulaire par adapté au problème du suivi. Une extension de cette approche est proposée dans [33]. Afin d'améliorer le suivi à l'aide du filtrage particulaire, l'espace d'état est partitionné. [103] utilise un filtrage de Kalman étendu pour estimer les paramètres de pose. Le vecteur d'état est donc l'ensemble des paramètres de pose. Etant donné la prédiction, une labelisation des voxels est effectuée et le taux de recouvrement des ellipsoïdes sur les voxels labellisés permet d'estimer une erreur à minimiser. L'algorithme met alors à jour les paramètres du modèle. [132] propose d'utiliser un *non-parametric belief propagation* avec une adaptation du filtrage particulaire pour effectuer l'estimation du mouvement. Un modèle de *graphs* du corps humain ainsi qu'un modèle d'évolution temporelle des différentes parties du corps sont appris en utilisant des données issues de *motion capture*. Pour effectuer l'estimation du mouvement, les auteurs proposent d'utiliser des détecteurs de parties du corps spécifiques et multi-images. Ces modèles permettent de détecter les différentes parties du corps dans les images et d'estimer ensuite les paramètres de pose en utilisant le modèle d'évolution temporelle.

Les méthodes par minimisation directe : L'objectif est de trouver l'ensemble des paramètres de pose du modèle qui minimisent la *distance* entre le modèle *3D* et ses images. De manière générale, ces approches prédisent la position des points d'intérêts, des contours, *etc.* dans les images, puis cherchent les trajectoires articulaires qui s'en approchent le plus. Cette classe généralise l'approche avec marqueurs et peut être résolue avec les mêmes outils. [19] (et plus tard dans [20]) propose d'utiliser un méthode de type Newton-Raphson pour minimiser la fonction d'erreur. Plus précisément, les auteurs proposent une méthode itérative pour estimer les paramètres de pose. La solution au moindre carré du problème est linéarisée autour de la solution. En utilisant la linéarisation et la solution calculée, la fonction de coût est réévaluée. De manière similaire, [34] utilise une approche dynamique pour effectuer le suivi. Les forces calculées entre les contours du modèle et les silhouettes dans les images sont estimées et les équations de la dynamique résolues. Les forces sont alors réévaluées avec les nouveaux paramètres de pose. Il s'agit d'un processus itératif de minimisation. Le critère d'arrêt est le faible mouvement du modèle lors de la résolution des équations. [39], tout comme [113], utilise un algorithme de type EM pour effectuer la capture du mouvement de la main (respectivement du corps humain). La phase d'Estimation est effectuée en utilisant un algorithme de type Levenberg Marquardt avec comme les d.d.l. de la chaîne cinématique comme paramètres.

2.3.7 Evaluation des résultats

Pour se rendre compte des évolutions des systèmes de capture du mouvement, il faut pouvoir définir un étalon de mesure et comparer les résultats de capture de mouvement sans marqueurs avec cet étalon. Ainsi, pour évaluer la qualité d'un système de capture du mouvement, nous pouvons comparer nos résultats à ceux donnés par un système industriel utilisant marqueurs. Cependant, peu de travaux proposent actuellement une telle comparaison. Nous avons eu la chance dans le projet SEMOCAP de disposer dans une même salle d'un système VICON à marqueurs et de notre système vidéo sans marqueurs afin d'effectuer des comparaisons. Notons que des bases données multi-caméras sont maintenant disponibles pour effectuer des études comparées de résultats. Nous pouvons citer la base MoBo du CMU (*Carnegie Mellon University*) ([62]) ou tout récemment la base HumanEva [133] mettant à disposition des données vidéo acquises en parallèle avec un système VICON.

Cependant, ce seul critère de précision n'est pas suffisant. En effet, l'absence de jeux de données issus de la capture du mouvement avec marqueurs n'empêche pas la possibilité d'évaluer les performances d'un système. Nous pouvons utiliser d'autres critères comme la robustesse (aux environnement de capture, la possibilité de détecter des échecs du suivi) et la vitesse d'exécution. Un dernier critère peut aussi être la facilité avec laquelle un utilisateur externe (mais expert) peut utiliser l'application, nous parlerons alors du degré d'automatisation du processus.

La précision : Les systèmes de capture du mouvement multi-caméras sans marqueurs sont développés afin de supplanter les systèmes avec marqueurs. Pour que ces der-

niers puissent être utilisés par des animateurs, il faut que la précision atteigne celle de systèmes industriels comme le VICON. Mais [56] montre que la capture du mouvement sans marqueurs est un véritable défi et que la précision requise par les animateurs pour la production de films ou de jeux n'est pas encore atteinte. En effet, l'utilisation de données image entraîne de manière intrinsèque des imprécisions. D'une part, la modélisation de l'acteur n'est pas fidèle à la réalité. D'autre part, l'extraction des données n'est jamais exacte. Bien évidement, avec les progrès actuels, l'augmentation de la puissance de calculs et l'amélioration de la qualité des images les approches utilisant la vidéo comme moyen de capturer le mouvement tendent à atteindre les besoins de précision que requièrent les animateurs.

Degré d'automatisation : L'utilisation de systèmes de capture du mouvement doit pouvoir être effectuée de manière quasi automatique avec très peu d'interventions d'un utilisateur externe. Cette automatisation comprend aussi bien l'étape de calibrage des caméras que l'initialisation du modèle *3D* que le traitement des séquences vidéos. Concernant le premier point (le calibrage des caméras), nous pouvons probablement dire qu'il s'agit d'un processus acquis et que des algorithmes de calibrage existent et sont performants. C'est en générale l'initialisation du modèle *3D* qui requiert le plus d'intervention humaine. Et c'est aussi de la précision de ce modèle que dépend la précision des résultats. Sur ce point, beaucoup de travaux mettent l'accent sur la possibilité d'initialiser de manière automatique ou semi-automatique le modèle. Les questions du dimensionnement et de l'initialisation sont traitées par [81], [52], [80], [33], [103], [134], [83] ou encore [138]. De manière générale, les approches semi-automatiques réclament de la part de l'utilisateur de sélectionner quelques articulations clefs dans les images pour pouvoir positionner le squelette. Puis le modèle d'apparence est créé de manière automatique à partir des données images. Nous avons décidé de prendre cette dernière approche. En effet, l'approche entièrement automatique de l'initialisation nécessite de contraindre l'acteur ou alors de supposer des heuristiques permettant de retrouver correctement la pose dans la première image. Des méthodes d'apprentissages ([2], [121]) peuvent permettre d'initialiser la pose de l'acteur de manière automatique.

Vitesse d'exécution : Il existe actuellement un compromis entre la vitesse d'exécution et la précision avec laquelle l'estimation du mouvement est effectuée. Les approches temps réel n'ont pas le même objectif que les approches qui peuvent être plus lentes. Certaines permettront une interaction homme-machine d'autre permettront de créer des bases de données de mouvement pour l'animation. Cependant, le temps de calcul reste un facteur important pour les animateurs. Lorsqu'une séquence est acquise, il est plus avantageux d'avoir les résultats le plus vite possible pour connaître la viabilité de la séquence. Nous pouvons donner quelques temps de calcul référencé dans certains travaux : une heure de traitement pour 5 sec. d'acquisition pour [33], temps réel ou 10 Hz. pour [45], 15 sec. par image pour [104], 0.9 sec par image [83], 6 minutes par image sous Matlab pour [10] ou encore 50 sec. par image pour [17]. Tous ces temps sont bien évidemment à mettre en

relation avec la date de parution des différents travaux. Nous avons adopté une approche rapide mais non temps réel, permettant d'obtenir le résultat de 5 sec. d'acquisition en dix minutes. L'absence de contrainte temps réel nous ouvre alors un large choix de méthodes pour effectuer le suivi du mouvement avec précision.

Robustesse : Ce critère est difficile à évaluer. Le temps moyen entre deux décrochages du suivi peut être un bon critère. Comment améliorer alors la robustesse des algorithmes ? Bien évidemment, nous pouvons faire moins d'erreur, mais nous pouvons aussi mieux diagnostiquer ces erreurs et en cas d'erreurs permettre à l'algorithme ou à un utilisateur de réinitialiser le suivi. Une autre solution est de réinitialiser le suivi de manière systématique à intervalles réguliers.

Contraintes d'acquisition : Ce dernier critère est plus un « plus » pour la capture du mouvement. De manière générale, les séances de capture du mouvement avec des systèmes vidéos ou des systèmes à marqueurs se passent dans des environnements très contrôlés. Cependant, l'idéal pour un système de capture du mouvement serait de pouvoir opérer dans n'importe quel environnement (un stade, un terrain de tennis, un bureau avec du monde). En effet, les studios spécialement conçus pour l'acquisition (studios bleus) sont chers à fabriquer ou très peu disponibles. Il s'agit de concevoir des systèmes qui soient opérationnels dans des environnements peu contrôlé et où l'acteur n'est pas tenu de porter une tenue vestimentaire particulière. Quelques approches multi-caméras comme celles de [45] ou encore de [119] proposent des résultats avec des scènes non contrôlées. Nous allons présenter dans cette thèse plusieurs approches permettant d'effectuer la capture du mouvement sans contraintes vestimentaires particulières pour l'acteur. Concernant, l'environnement de capture, nous verrons que nous nous autorisons un environnement lumineux peu contrôlé bien qu'il soit préférable qu'il le soit pour effectuer une soustraction de fond d'image utilisable pour l'estimation du mouvement.

Dans le projet SEMOCAP, nous verrons que toutes les contraintes devaient être prises en compte. En l'absence de données *test*, il est très difficile d'évaluer et de comparer les méthodes existantes. Espérons que cela devienne possible à l'avenir avec les base comme celle du CMU ou de BROWN. Mais cela représentera un travail considérable.

Première partie

Analyse et capture du mouvement humain

Chapitre 3

Rappels : modélisation des mouvements articulés

Sommaire

Résumé

Dans ce chapitre, nous introduisons une méthodologie empruntée à la robotique pour paramétrer une chaîne articulaire complexe (le corps humain) à l'aide de *paramètres articulaires* naturels et bien définis. Ces paramètres forment un vecteur $\mathbf{\Phi}$ qui permet d'exprimer le torseur cinématique de chacun des éléments de la chaîne articulaire sous la forme :

$$\begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{pmatrix} = \mathbf{J}_H \dot{\mathbf{\Phi}}, \tag{3.1}$$

où $\mathbf{J}_H$ est la Jacobienne du mouvement articulaire de la chaîne. Cette expression, nous permet d'exprimer le mouvement d'un point $\boldsymbol{X}$ quelconque de la chaîne articulaire par une formule simple :

$$\begin{aligned} \dot{\boldsymbol{X}} &= \boldsymbol{v} + \boldsymbol{\omega} \times (\boldsymbol{X} - \boldsymbol{t}) && (3.2) \\ &= \begin{bmatrix} [\boldsymbol{t} - \boldsymbol{X}]_\times & \mathbf{I} \end{bmatrix} \mathbf{J}_H \dot{\mathbf{\Phi}}. && (3.3) \end{aligned}$$

Pour aboutir à cette formulation, nous rappelons dans une première partie de ce chapitre les bases pour exprimer la position, la rotation et le mouvement d'un point rigidement attaché à un référentiel mobile et ce par rapport à un référentiel fixe. A partir de ces rappels, nous étendons la modélisation au cas d'une chaîne cinématique ouverte (c'est-à-dire sans cycles). Nous exprimons alors la position et la vitesse d'un point rigidement attaché à l'un des éléments de la chaîne en fonction des paramètres articulaires (de la chaîne). Plus précisément, nous donnons l'expression analytique de $\mathbf{J}_H$.

Ces rappels de cinématique et la modélisation de la chaîne cinématique proposés dans ce chapitre nous permettront de modéliser le squelette du corps humain (dans le chapitre 4). Pour effectuer le suivi du mouvement, nous utiliserons le formalisme introduit dans ce chapitre. Nous pourrons alors expliciter le mouvement d'un point du corps humain en fonction de la variation des paramètres de pose de l'acteur.

Introduction au chapitre

Afin de modéliser le corps humain et d'effectuer la capture du mouvement, nous devons modéliser la chaîne cinématique que constitue le squelette humain. Plus précisément, nous devons modéliser le mouvement d'un point attaché à l'un des éléments de la chaîne cinématique. La position et le mouvement d'un élément de la chaîne cinématique peuvent être abordé de deux points de vue :

- Nous pouvons paramétrer le mouvement d'un élément de la chaîne à l'aide de 3 rotations et 3 translations. Nous parlons alors du mouvement rigide d'un objet.
- Nous pouvons exprimer le mouvement en fonction de tous les degrés de liberté (d.d.l.) de la chaîne cinématique. Le mouvement est alors exprimé à l'aide des paramètres de pose de la chaîne.

Ces deux point de vue ne sont pas indépendant. En effet, nous pouvons exprimer le mouvement libre en fonction des paramètres de pose de la chaîne.

Dans un premier temps, nous motiverons ce chapitre en explicitant le principe du suivi du mouvement. Puis, nous rappellerons les différents formalismes utilisés pour décrire les rotations ainsi que le mouvement rigide d'un objet dans l'espace *3D*. Nous justifierons alors les choix de modélisation que nous avons adoptés dans cette thèse. Nous aborderons ensuite la modélisation d'une chaîne cinématique. Nous introduirons alors la modélisation en référence zéro permettant d'exprimer le déplacement d'un point de la chaîne cinématique entre deux positions. Enfin, nous donnerons l'expression analytique du mouvement d'un point de la chaîne en fonction des variations des paramètres de pose.

3.1 Motivations

Dans ce chapitre, nous nous attardons sur la modélisation du mouvement d'un point attaché à un élément d'une chaîne cinématique. Plus précisément, nous développons le formalisme mathématique nécessaire pour exprimer la Jacobienne d'une chaîne cinématique liant la variation des paramètres articulaires de la chaîne au torseur cinématique associé à un point de la chaîne. Dans cette section, nous allons montrer comment intervient cette modélisation dans le processus d'estimation du mouvement.

Considérons une méthode d'estimation du mouvement utilisant des marqueurs (comme le système VICON décrit dans le chapitre 2 à la section 2.1). Supposons que les marqueurs détectés dans les images sont reconstruits en *3D*. Nous allons aborder la phase d'estimation des paramètres de pose du modèle *3D*.

Soit $\boldsymbol{X}_{obs}^{\mathcal{W}}(m)$ les coordonnées *3D* du marqueur m observé exprimées dans le repère du monde et $\boldsymbol{X}_{t}^{\mathcal{W}}(m)$ les coordonnées à l'instant t du marqueur modèle correspondant exprimées dans le repère du monde.

L'objectif de la capture du mouvement avec marqueurs est d'estimer les paramètres

du modèle *3D*, tels que :

$$E = \sum_m \|\boldsymbol{X}_{obs}^{\mathcal{W}} - \boldsymbol{X}_t^{\mathcal{W}}(m)\|^2 = 0. \tag{3.4}$$

$\boldsymbol{X}_t^{\mathcal{W}}$ dépend des paramètres de pose du modèle. En notant $\boldsymbol{\Phi}$ le vecteur des paramètres, nous avons :

$$\boldsymbol{X}_t^{\mathcal{W}}(m) = f(\boldsymbol{X}_0^{\mathcal{W}}(m), \boldsymbol{\Phi}), \tag{3.5}$$

où $\boldsymbol{X}_0^{\mathcal{W}}(m)$ est la position de référence du marqueur. En dérivant, nous avons :

$$\dot{\boldsymbol{X}}_t^{\mathcal{W}}(m) = \mathbf{J}_f(\boldsymbol{X}_0^{\mathcal{W}}(m), \boldsymbol{\Phi})\dot{\boldsymbol{\Phi}}, \tag{3.6}$$

où $\mathbf{J}_f$ est la Jacobienne de la fonction f. Cette équation permet d'obtenir une solution itérative au problème de cinématique inverse permettant de résoudre l'équation (3.4), avec pour chaque marqueur :

$$\boldsymbol{X}_{k+1}^{\mathcal{W}} = \boldsymbol{X}_k^{\mathcal{W}} + \mathbf{J}(\boldsymbol{X}_0^{\mathcal{W}}, \boldsymbol{\Phi})\Delta\boldsymbol{\Phi}, \tag{3.7}$$

où k est l'itération pour la résolution du système (à l'instant t de la séquence vidéo). L'objectif est de calculer $\Delta\boldsymbol{\Phi}$ tel que $\boldsymbol{X}_{k+1} = \boldsymbol{X}_{obs}^{\mathcal{W}}$, nous avons donc :

$$\Delta\boldsymbol{\Phi} = (\mathbf{J}\mathbf{J}^\top)^{-1}\mathbf{J}^\top(\boldsymbol{X}_{obs}^{\mathcal{W}} - \boldsymbol{X}_t^k). \tag{3.8}$$

Tout au long de ce chapitre, nous allons développer les outils nécessaires pour exprimer $\mathbf{J}$. Plus précisément, nous verrons que :

$$\dot{\boldsymbol{X}}^{\mathcal{W}} = \begin{bmatrix} [\boldsymbol{t} - \boldsymbol{X}^{\mathcal{W}}]_\times & \mathbf{I}_{3\times3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}, \tag{3.9}$$

où $\boldsymbol{t}$ est la position du centre du repère associé à $\boldsymbol{X}^{\mathcal{M}}$ et

$$\begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} = \mathbf{J}_H\dot{\boldsymbol{\Phi}} \tag{3.10}$$

est le torseur cinématique associé au point avec $\mathbf{J}_H$ qui sera explicité à la fin de ce chapitre.

Nous avons développé l'exemple de l'estimation du mouvement avec marqueurs. Dans le cadre de cette thèse, nous n'utilisons pas de marqueurs. Pour représenter l'acteur, nous utilisons un modèle *3D* que nous mettons en correspondance avec les images. Pour effectuer cette mise en correspondance, nous projetons le modèle *3D* dans chacune des images. Dans le chapitre 4, nous explicitons le modèle du corps humain et nous complétons les équations introduites dans ce chapitre de façon à relier le mouvement apparent du modèle dans les images avec les variations des paramètres articulaires *3D*. Nous y étudierons le cas du *contour apparent* (contour du modèle projeté dans les images) en détails. Nous verrons alors que toutes les observations, y compris les contours, peuvent être intégrées à une formule similaire à 3.4 et résolues par une formule similaire à 3.8.

3.2 Les rotations

L'ensemble des rotations dans $\mathbb{R}^3$ forme un groupe dénoté $SO(3)$ (pour *special orthogonal*). En notant $\mathbf{R}$ une matrice de rotation, alors la notion de SO implique que $\det\mathbf{R} = +1$, où det est le déterminant d'une matrice. D'autre part, l'ensemble des rotations décrit un groupe de Lie. Les rotations sont donc continues et différentiables. Nous utiliserons cette propriété pour établir la vitesse de déplacement d'un objet rigide.

Dans la suite, nous considérons deux repères $\mathcal{F}$ (repère fixe) et $\mathcal{M}$ (repère mobile), ayant une origine commune. Notons $\boldsymbol{X}^{\mathcal{F}}$ ($\boldsymbol{X}^{\mathcal{M}}$) les coordonnées du point X exprimées dans le référentiel $\mathcal{F}$ (respectivement $\mathcal{M}$). Si nous notons $\mathbf{R}$ l'orientation de $\mathcal{M}$ par rapport à $\mathcal{F}$, alors :

$$\boldsymbol{X}^{\mathcal{F}} = \mathbf{R}\boldsymbol{X}^{\mathcal{M}}. \tag{3.11}$$

Dans l'espace *3D* ($\mathbb{R}^3$), les rotations peuvent être représentées de différentes manières. Nous présentons dans un premier temps la représentation exponentielle encore appelée la représentation dite de *twist*. Puis nous abordons rapidement la description avec les quaternions. Ensuite, nous nous attardons sur la représentation à l'aide des angles d'Euler. Puis, nous discutons des principaux avantages et inconvénients de chacune des représentations pour enfin expliciter et justifier nos choix.

3.2.1 Représentation exponentielle

Supposons deux repères orthonormés initialement confondus $\mathcal{F} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ et $\mathcal{M} = (\boldsymbol{e}_1^{'}, \boldsymbol{e}_2^{'}, \boldsymbol{e}_3^{'})$. Un point X a pour coordonnées :

$$\boldsymbol{X} = \sum_{i=0}^{3} x_i \boldsymbol{e}_i = \sum_{i=0}^{3} x_i \boldsymbol{e}_i^{'}.$$

Supposons X rigidement attaché à $\mathcal{M}$. Nous pouvons exprimer la vitesse de $\boldsymbol{X}$ dans $\mathcal{F}$:

$$\dot{\boldsymbol{X}} = \sum_{i=0}^{3} x_i \frac{d\boldsymbol{e}_i^{'}}{dt} = \sum_{i=0}^{3} x_i \sum_{j=0}^{3} \omega_{ij} \boldsymbol{e}_j. \tag{3.12}$$

En réordonnant les termes, nous avons le produit d'une matrice $\boldsymbol{\Omega} = (\omega_{i,j})_{i,j\in[0..2]}$ et du vecteur $\boldsymbol{X}$:

$$\dot{\boldsymbol{X}} = \boldsymbol{\Omega}\boldsymbol{X}. \tag{3.13}$$

D'autre part, $\boldsymbol{\Omega}$ est une matrice antisymétrique. En effet, par définition d'une base orthornormée, nous avons $\boldsymbol{e}_i \cdot \boldsymbol{e}_j = 0$. En dérivant cette égalité et en utilisant les termes ω_{ij} introduits précédemment, nous avons $w_{ij} + w_{ji} = 0$. Nous avons donc $\omega_{ij} = -\omega_{ji}$. $\boldsymbol{\Omega}$ est donc bien une matrice antisymétrique. Nous pouvons donc écrire l'équation (3.13) comme un produit vectoriel :

$$\dot{\boldsymbol{X}} = \boldsymbol{\omega} \times \boldsymbol{X}, \tag{3.14}$$

où $\mathbf{\Omega} = [\boldsymbol{\omega}]_\times$. La notation $[\boldsymbol{\omega}]_\times$ définit la matrice antisymétrique construite à partir du vecteur $\boldsymbol{\omega}$. En posant $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$, alors :

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3.15}$$

En notant $\omega = \|\boldsymbol{\omega}\|$, alors ω est la vitesse de rotation associé à $\mathbf{\Omega}$ et $\omega' = \omega^{-1}\boldsymbol{\omega}$ est l'axe instantané de rotation. $\boldsymbol{\omega}$ est le **vecteur de vitesse en rotation**.

Nous pouvons maintenant exprimer la rotation en fonction de $\boldsymbol{\omega}$. Si $\boldsymbol{\omega}'$ est fixe au cours du temps, on peut aisément résoudre l'équation différentielle (3.13) qui a pour solution :

$$\boldsymbol{X} = e^{\boldsymbol{\omega}'\omega t}\boldsymbol{X}_0. \tag{3.16}$$

Or $e^{\boldsymbol{\omega}'\omega t}$ est une matrice de rotation. En effet, en posant $\omega t = \theta$, nous pouvons écrire (c.f. annexe A) :

$$\boxed{e^{[\boldsymbol{\omega}]_\times\theta} = \mathbf{I} + \sin\theta[\boldsymbol{\omega}]_\times + (1-\cos\theta)[\boldsymbol{\omega}]_\times^2}, \tag{3.17}$$

qui a bien toutes les propriétés d'une matrice de rotation. L'équation (3.17) est connue sous le nom de formule de Rodrigues.

Nous pouvons donc poser :

$$\mathbf{R}(\boldsymbol{\omega}, \theta) = e^{[\boldsymbol{\omega}]_\times\theta}, \tag{3.18}$$

avec, par abus de notation, $\boldsymbol{\omega} = \boldsymbol{\omega}'$.

Enfin, nous pouvons inverser l'équation (3.17) afin de déterminer l'axe instantané et l'angle de la rotation :

$$\rightarrow \theta = \cos^{-1}\left(\frac{tr(\mathbf{R})-1}{2}\right), \tag{3.19}$$

$$\rightarrow \boldsymbol{\omega} = \frac{1}{2\sin\theta}\begin{bmatrix} m_{32}-m_{23} \\ m_{13}-m_{31} \\ m_{21}-m_{12} \end{bmatrix}, \tag{3.20}$$

avec $\theta \neq 0$, $\mathbf{R} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$ et $tr(\mathbf{R})$ la trace de $\mathbf{R}$.

Si $\theta = 0$ alors $\boldsymbol{\omega}$ peut être choisi arbitrairement. Il s'agit là d'une singularité. La représentation exponentielle n'est pas bijective. Pour toute rotation donnée (axe et angle) il existe une représentation exponentielle unique. Mais pour une représentation exponentielle donnée, il peut exister plusieurs choix possibles pour l'axe et l'angle.

3.2.2 Les quaternions

Les quaternions généralisent les rotations exprimées dans le plan sous forme complexe.

Dans l'espace des quaternions, les rotations sont représentées par un quadruplé de nombre réels (a, b, c, d) : un scalaire plus un vecteur de $\mathbb{R}^3$. Un quaternion $\boldsymbol{q}$ se met sous la forme : $\boldsymbol{q} = a.1 + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, où $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$ sont des nombres complexes purs. [46] propose d'introduire les quaternions pour modéliser les rotations. Leur objectif était de palier les singularités liées aux angles d'Euler (c.f. 3.2.3). [70] propose d'utiliser les quaternions pour modéliser l'espace des contraintes de déplacement des articulations. Shoemake propose d'utiliser ces quaternions pour effectuer de l'interpolation de mouvement en animation ([130]). En effet, l'utilisation des quaternions permet de simplifier les calculs de composition de matrices.

Nous pouvons faire le lien entre cette représentation est celle que nous avons exposé ci-dessus. En effet, un quaternion $\boldsymbol{q}$ s'écrit sous la forme :

$$\boldsymbol{q} = \cos\theta/2 + \sin\theta/2(i\omega_1 + j\omega_2 + k\omega_3), \tag{3.21}$$

où θ est l'angle de rotation et $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^\top$ l'axe instantané de rotation.

Cette représentation présente l'avantage de ne pas avoir de singularités contrairement à la représentation exponentielle. Cependant nous n'utiliserons pas cette représentation pour la modélisation de notre squelette (c.f. la discussion dans le paragraphe 3.3.3).

3.2.3 Les angles d'Euler

Une autre façon classique de représenter une rotation quelconque est de privilégier les rotations autour des axes des deux repères $\mathcal{M}$ et $\mathcal{F}$. La rotation de $\mathcal{M}$ par rapport à $\mathcal{F}$ est exprimée en utilisant 3 angles (Θ, Φ et Ψ). Ce triplé permet de construire une matrice de rotation permettant de passer des coordonnées d'un point exprimé dans $\mathcal{M}$ aux coordonnées exprimées dans $\mathcal{F}$. Cependant, il existe plusieurs conventions pour la construction de la matrice : la convention d'axe fixe, d'axes en mouvement et la convention d'angle fixe. Les conventions d'Euler sont une composition de trois rotations autour de deux axes. De manière générale, nous notons les ordres de rotation de la manière suivante : KJK (noté en majuscule pour notifié les rotations autour des axes fixes) ou encore kjk (noté en minuscule pour notifier les rotations autour des axes mobiles). Pour plus de détails sur les conventions d'Euler, le lecteur pourra se référer à l'annexe A.2. Cependant, nous utiliserons la convention de Cardan, souvent confondue avec celle d'Euler, qui est du type IJK, pour laquelle les rotations se font autour des trois axes.

Remarque : Les trois conventions des angles d'Euler sont équivalentes. De même, si nous considérons les angles de Cardan, nous avons équivalence entre la convention d'axe fixe et la convention d'axe en mouvement. Nous pouvons montrer, par exemple,

que $\mathbf{R}_{KJI}(\Theta, \Phi, \Psi) = \mathbf{R}_{ijk}(\Psi, \Phi, \Theta)$ (notez l'inversion de l'ordre dans les angles). La preuve d'une des équivalences est donnée dans l'annexe A.3.3.

Pour certaines applications, il peut être nécessaire de calculer les angles de rotations à partir des matrices de rotation. Dans le cadre de nos travaux, ce calcul est nécessaire pour effectuer des conversions entre les différents formats de données que nous utilisons. Différentes conventions d'orientation des repères sont utilisées dans les différentes applications que nous utilisons. Pour effectuer la conversion des données, nous utilisons la matrice de rotation qui reste identique pour les différentes conventions. Seules les valeurs des angles d'Euler extraits de la matrice changent selon la convention choisie. Le passage d'une convention à l'autre nécessite alors l'extraction des angles à partir des matrices de rotations. Nous n'aborderons pas, ici, le calcul des angles, cependant le lecteur pourra se référer à l'annexe A.3 où nous décrivons la méthode ainsi que l'implémentation utilisée.

Singularités ou effet *Gimbal Lock* Les angles d'Euler permettent de décrire toutes rotations dans $SO(3)$ de manière unique. Cependant une rotation de $SO(3)$ peut ne pas avoir de solution unique pour les angles d'Euler. Par exemple, pour le formalisme de type KJK, tout triplet de la forme $(\Theta, 0, -\Theta)$ donne pour matrice $\mathbf{R}_{KJK} = \mathbf{I}$ où $\mathbf{I}$ est la matrice identité (de dimension 3×3). Il s'agit donc d'une singularité car l'extraction des angles à partir de la matrice identité n'a pas de solution unique. Contrairement aux conventions d'Euler, la représentation dite de Cardan ne souffre pas de cette singularité pour la matrice $\mathbf{I}$. Cependant, ces singularités existent lorsque, par exemple pour la convention kji, $\Phi = -\pi/2$. Ces singularités sont connues sous le nom de *gimbal lock*. Elles entraînent notamment la perte d'un d.d.l. dans le mouvement. Plusieurs solutions pratiques peuvent être envisagées pour éviter de tomber dans ce type de singularité. Nous en développerons quelques-unes dans le chapitre 4 au paragraphe sur la modélisation de la chaîne cinématique du corps humain (paragraphe 4.1).

3.3 Le déplacement rigide

Nous avons développé le formalisme concernant les mouvements de rotation. Nous allons maintenant aborder le déplacement rigide qui est la composition d'une rotation et d'une translation.

Reconsidérons les deux référentiels $\mathcal{F}$ et $\mathcal{M}$ introduits précédemment. Nous nous plaçons dans le cas où leurs origines (O_1 et O_2) ne sont plus confondues. Nous pouvons définir le vecteur $\boldsymbol{t} = \overrightarrow{O_1O_2}$. De même que précédemment, nous définissons la matrice $\mathbf{R}$, la rotation de $\mathcal{M}$ par rapport à $\mathcal{F}$. Le couple $(\mathbf{R}, \boldsymbol{t})$ définit une configuration de $\mathcal{M}$ par rapport à $\mathcal{F}$. On définit $SE(3)$ (pour *Special Euclidian*) l'ensemble des configurations telles que $\boldsymbol{t} \in \mathbb{R}^3$ et $\mathbf{R} \in SO(3)$. Tout comme $SO(3)$, $SE(3)$ est un groupe et définit un groupe de Lie.

Tout point X dont les coordonnées sont exprimées dans $\mathcal{M}$ a pour coordonnées dans $\mathcal{F}$:

$$\boldsymbol{X}^{\mathcal{F}} = \mathbf{R}\boldsymbol{X}^{\mathcal{M}} + \boldsymbol{t}. \tag{3.22}$$

Comme pour les rotations, il y a plusieurs représentations possibles pour exprimer ce changement de référentiel. Nous aborderons ici rapidement la représentation matricielle ainsi que la représentation exponentielle. Puis nous donnerons les relations permettant d'exprimer le déplacement d'un objet rigide : c'est-à-dire l'expression de la position d'un objet à un instant donné par rapport à un référentiel donné en fonction de sa position antérieure (dans ce même référentiel).

3.3.1 Représentation matricielle du changement de repère

Si nous adoptons les coordonnées homogènes, la transformation (3.22) peut s'exprimer de la manière suivante :

$$\overline{\boldsymbol{X}}^{\mathcal{F}} = \mathbf{M}_{\mathcal{MF}}\overline{\boldsymbol{X}}^{\mathcal{M}}, \tag{3.23}$$

où $\overline{\boldsymbol{X}} = (\boldsymbol{X}, 1)^\top$ est la représentation en coordonnées homogènes du vecteur $\boldsymbol{X}$. $\mathbf{M}_{\mathcal{MF}}$ est une matrice de dimension 4×4 représentant le changement de référentiel de $\mathcal{M}$ vers $\mathcal{F}$. $\mathbf{M}_{\mathcal{MF}}$ est de la forme :

$$\mathbf{M}_{\mathcal{MF}} = \begin{bmatrix} \mathbf{R} & \boldsymbol{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{3.24}$$

De la même manière, en inversant l'équation (3.22), nous pouvons exprimer $\boldsymbol{X}^{\mathcal{F}}$ dans le référentiel $\mathcal{M}$:

$$\boldsymbol{X}^{\mathcal{M}} = \mathbf{R}^{-1}(\boldsymbol{X}^{\mathcal{F}} - \boldsymbol{t}) = \mathbf{R}^\top(\boldsymbol{X}^{\mathcal{F}} - \boldsymbol{t}). \tag{3.25}$$

Nous avons alors :

$$\mathbf{M}_{\mathcal{MF}}^{-1} = \mathbf{M}_{\mathcal{FM}} = \begin{bmatrix} \mathbf{R}^\top & -\mathbf{R}^\top\boldsymbol{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{3.26}$$

Cette représentation compacte des changements de référentiel permet de chaîner les transformations. Il suffit alors de multiplier successivement les matrices de passage. Nous utiliserons ce chaînage pour la chaîne cinématique.

3.3.2 Représentation exponentielle

La représentation exponentielle introduite précédemment dans le cadre des rotations (paragraphe 3.2.1) est généralisable aux transformations rigides générales. Nous n'aborderons pas cette généralisation ici. Le lecteur pourra se référer à [109] (p. 39–50). En pratique, nous utiliserons la représentation exponentielle uniquement pour les rotations.

3.3.3 Choix et Discussion

Nous avons modélisé dans ce début de chapitre, la position et la rotation d'un objet rigide. Nous verrons que nous utiliserons cette modélisation dans le cadre d'objets articulés et notamment pour modéliser la chaîne cinématique du corps humain. Le

corps humain a des contraintes anatomiques qu'il faut pouvoir modéliser à l'aide de la représentation choisie.

Pour effectuer le choix de la modélisation, nous devons donc prendre en compte deux types de considérations :

- Le choix de paramètres doit être général et donc pouvoir être adapté pour la description de toute chaîne articulée,
- Le choix doit permettre d'exprimer aisément les contraintes anatomiques. Notamment, nous devons pouvoir bloquer certains degrés de rotations pour certaines articulations.

Nous avons présenté plusieurs représentations possibles pour exprimer la position et l'orientation d'un objet. Concernant l'expression de la position, le choix n'est pas à faire. Il s'agit de la modéliser par un vecteur à trois composantes. Nous allons donc présenter les avantages et inconvénients des différentes modélisations de la rotation que sont la forme exponentielle, les quaternions et les angles d'Euler. Nous finirons ce paragraphe en donnant la représentation que nous avons adoptée pour cette thèse.

L'estimation des paramètres L'estimation des paramètres de pose d'un objet articulé est nécessaire pour effectuer le suivi de l'acteur. Pour chacune des représentations exposées, il s'agit d'estimer à chaque instant l'ensemble des paramètres la caractérisant :

- trois angles pour la représentation d'Euler ou de Cardan,
- un vecteur unitaire et un angle, soit quatre paramètres, pour les quaternions,
- un axe et un angle, soit quatre paramètres, pour la représentation exponentielle.

L'estimation de la rotation à l'aide des quaternions nécessite de vérifier la contrainte unitaire sur le quadruplé estimé : $\|\boldsymbol{q}\|^2 = 1$. Cette contrainte est satisfaite en estimant trois des paramètres des quaternions et en déduisant le quatrième.

La représentation sous forme exponentielle souffre de singularités pour des angles de rotations instantanés nuls. Cependant, elle a l'avantage de ne plus avoir la contrainte de vecteur unitaire pour $\boldsymbol{\omega}$. Cette représentation semble donc la plus adéquate pour la modélisation des rotations. La représentation d'Euler ne compte que trois paramètres à estimer, cependant, comme nous l'avons souligné, cette représentation comporte également des singularités (effet de *Gimbal Lock*).

Les contraintes articulaires La modélisation de la chaîne cinématique nécessite de pouvoir restreindre le nombre de d.d.l. pour les rotations. Prenons par exemple le cas du genou, où dans une première approximation, l'articulation ne comporte qu'un seul d.d.l. En représentation d'Euler, cela signifie que deux des trois angles doivent être maintenus à valeurs constantes. Cette contrainte est plus difficile à traduire pour la représentation exponentielle ou pour la représentation sous forme de quaternions. En effet, Usta dans [145] effectue une étude comparative de l'utilisation des représentations d'Euler et des quaternions pour modéliser la chaîne cinématique humaine. La conclusion donnée reste mitigée sur l'avantage de l'utilisation d'une représentation par rapport à l'autre. Cependant, bien que les angles d'Euler aient des singularités, leur utilisation est plus efficace que celle des quaternions. En effet, pour contraindre les rotations pour

les quaternions (sans prè-apprentissage comme dans [70]), il est nécessaire d'effectuer une conversion vers les angles d'Euler. Nous n'avons donc pas d'avantages à utiliser la représentation sous forme de quaternions. La représentation exponentielle permet de modéliser un, deux ou trois d.d.l., mais cela nécessite des multiplications matricielles supplémentaires. Nous le verrons dans la partie concernant la modélisation de la chaîne cinématique.

Malgré l'effet de *Gimbal Lock*, nous avons choisi d'utiliser la représentation sous forme d'angles de Cardan pour effectuer en pratique l'estimation des paramètres. Cependant, dans le paragraphe suivant, nous verrons qu'il est plus compact de représenter une chaîne cinématique à l'aide de la forme exponentielle. Nous développerons donc tout le formalisme avec cette dernière représentation bien qu'en pratique l'implémentation soit effectuée avec les angles de Cardan.

3.4 Paramétrage du mouvement rigide

Dans les paragraphes précédents, nous avons modélisé la position et la rotation d'un point ou de manière équivalente d'un objet rigide. Nous allons maintenant considérer le mouvement d'un point rigidement attaché à un objet que nous allons paramétrer à l'aide des angles d'Euler. Cette modélisation dynamique est nécessaire pour effectuer le suivi du mouvement. Nous allons donc dériver les différentes modélisations exposées précédemment pour expliciter analytiquement la vitesse d'un objet dans un référentiel donné. Nous commencerons par aborder le cas d'un mouvement de rotation pure pour ensuite aborder le cas plus général du mouvement rigide.

3.4.1 La vitesse de rotation

Nous avons vu qu'une rotation pouvait être exprimée à partir des angles d'Euler (ou de Cardan), de la représentation exponentielle ou d'un quaternion. Nous développons ici la dérivation des deux premières représentations en montrant l'avantage de la représentation exponentielle. Le lecteur pourra se référer à l'annexe A.4 pour la dérivation des quaternions.

Reconsidérons les deux repères $\mathcal{F}$ et $\mathcal{M}$, $\mathbf{R}$ la rotation de $\mathcal{M}$ par rapport à $\mathcal{F}$ et un point X rigidement attaché à $\mathcal{M}$. Dans un premier temps, nous considérons que les origines des deux repères sont confondues ($O_1 = O_2$).

Avec l'équation (3.11), nous avons :

$$\boldsymbol{X}^{\mathcal{F}} = \mathbf{R}\boldsymbol{X}^{\mathcal{M}} = \mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)\mathbf{R}_k(\Theta)\boldsymbol{X}^{\mathcal{M}}, \tag{3.27}$$

en utilisant la convention de Cardan.

La vitesse du point X dans le référentiel fixe est :

$$\dot{\boldsymbol{X}}^{\mathcal{F}} = \dot{\mathbf{R}}\boldsymbol{X}^{\mathcal{M}} + \mathbf{R}\dot{\boldsymbol{X}}^{\mathcal{M}} = \dot{\mathbf{R}}\boldsymbol{X}^{\mathcal{M}}, \tag{3.28}$$

car X est un point fixe dans $\mathcal{M}$ et sa vitesse est nulle dans ce référentiel ($\dot{\boldsymbol{X}}^{\mathcal{M}} = 0$).

Si nous utilisons le formalisme de Cardan pour représenter la rotation, nous avons :

$$\begin{aligned} \mathbf{R} &= \mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)\mathbf{R}_k(\Theta), & (3.29)\\ \mathbf{R}^\top &= \mathbf{R}_k(-\Theta)\mathbf{R}_j(-\Phi)\mathbf{R}_i(-\Psi). & (3.30) \end{aligned}$$

La dérivée de $\mathbf{R}$ est de la forme :

$$\dot{\mathbf{R}} = \dot{\mathbf{R}}_i(\Psi)\mathbf{R}_j(\Phi)\mathbf{R}_k(\Theta) + \mathbf{R}_i(\Psi)\dot{\mathbf{R}}_j(\Phi)\mathbf{R}_k(\Theta) + \mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)\dot{\mathbf{R}}_k(\Theta), \tag{3.31}$$

avec :

$$\dot{\mathbf{R}}_i(\Psi) = \dot{\Psi}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -s(\Psi) & -c(\Psi) \\ 0 & c(\Psi) & -s(\Psi) \end{bmatrix}, \quad \dot{\mathbf{R}}_j(\Phi) = \dot{\Phi}\begin{bmatrix} -s(\Phi) & 0 & c(\Phi) \\ 0 & 0 & 0 \\ -c(\Psi) & 0 & -s(\Psi) \end{bmatrix} \text{ et}$$

$$\dot{\mathbf{R}}_k(\Theta) = \dot{\Theta}\begin{bmatrix} -s(\Theta) & -c(\Theta) & 0 \\ c(\Theta) & -s(\Theta) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \text{où } c = \cos \text{ et } s = \sin.$$

D'un point de vue théorique, la modélisation du mouvement à l'aide des angles d'Euler ne permet pas de manipuler facilement les expressions. Nous allons donc utiliser et modifier l'équation (3.28) pour obtenir une représentation compacte de la vitesse de rotation. Pour exprimer cette rotation, nous allons définir l'opérateur tangent de la rotation dont nous déduirons l'expression formelle de la matrice $\boldsymbol{\Omega}$ introduite au paragraphe 3.2.1.

L'opérateur tangent En utilisant l'équation (3.11), nous pouvons exprimer $\boldsymbol{X}^{\mathcal{F}}$ dans le référentiel $\mathcal{M}$:

$$\boldsymbol{X}^{\mathcal{M}} = \mathbf{R}^{-1}\boldsymbol{X}^{\mathcal{F}}. \tag{3.32}$$

Nous pouvons alors réécrire l'équation (3.28) de la manière suivante :

$$\dot{\boldsymbol{X}}^{\mathcal{F}} = \dot{\mathbf{R}}\mathbf{R}^{-1}\boldsymbol{X}^{\mathcal{F}}. \tag{3.33}$$

Calculons $\dot{\mathbf{R}}\mathbf{R}^{-1}$ à l'aide des équations (3.30) et (3.31). Nous remarquons qu'en multipliant ces deux équations, le premier terme se simplifie :

$$\dot{\mathbf{R}}_i(\Psi)\mathbf{R}_j(\Phi)\mathbf{R}_k(\Theta)\mathbf{R}_k(-\Theta)\mathbf{R}_j(-\Phi)\mathbf{R}_i(-\Psi) = \dot{\mathbf{R}}_i(\Psi)\mathbf{R}_i(-\Psi). \tag{3.34}$$

Nous pouvons faire de même avec les autre termes, nous obtenons donc l'expression suivante :

$$\begin{aligned} \dot{\mathbf{R}}\mathbf{R}^{-1} &= \dot{\mathbf{R}}_i(\Psi)\mathbf{R}_i^{-1}(\Psi) \\ &\quad +\mathbf{R}_i(\Psi)\dot{\mathbf{R}}_j(\Phi)\mathbf{R}_j^{-1}(\Phi)\mathbf{R}_i^{-1}(\Psi) \\ &\quad +\mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)\dot{\mathbf{R}}_k(\Theta)\mathbf{R}_k^{-1}(\Theta)\mathbf{R}_j^{-1}(\Phi)\dot{\mathbf{R}}_i^{-1}(\Psi). \end{aligned} \tag{3.35}$$

Si nous prenons par exemple le premier terme de la somme, nous avons :

$$\begin{aligned}
\dot{\mathbf{R}}_i(\Psi)\mathbf{R}_i^{-1}(\Psi) &= \dot{\Psi}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -s(\Psi) & -c(\Psi) \\ 0 & c(\Psi) & -s(\Psi) \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\Psi) & s(\Psi) \\ 0 & -s(\Psi) & c(\Psi) \end{bmatrix} \\
&= \dot{\Psi}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (3.36) \\
&= \dot{\Psi}[\boldsymbol{e}_i]_\times, \qquad (3.37)
\end{aligned}$$

où $\boldsymbol{e}_i = (1,0,0)^\top$. Nous avons donc, par extension :

$$\begin{aligned}
\dot{\mathbf{R}}\mathbf{R}^{-1} &= \dot{\Psi}[\boldsymbol{e}_i]_\times \\
&\quad +\dot{\Phi}\mathbf{R}_i(\Psi)[\boldsymbol{e}_j]_\times\mathbf{R}_i^{-1}(\Psi) \\
&\quad +\dot{\Theta}\mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)[\boldsymbol{e}_k]_\times\mathbf{R}_j^{-1}(\Phi)\mathbf{R}_i^{-1}(\Psi) \qquad (3.38) \\
&= [\boldsymbol{\omega}]_\times, \qquad (3.39)
\end{aligned}$$

où $\boldsymbol{e}_j = (0,1,0)^\top$ et $\boldsymbol{e}_k = (0,0,1)^\top$. $\dot{\mathbf{R}}\mathbf{R}^{-1}$ est appelé l'**opérateur tangent** de la matrice $\mathbf{R}$ et est noté $\widehat{\mathbf{R}}$.

L'équation (3.33) peut donc s'écrire sous la forme :

$$\dot{\boldsymbol{X}}^{\mathcal{F}} = [\boldsymbol{\omega}]_\times \boldsymbol{X}^{\mathcal{F}}. \qquad (3.40)$$

Nous retrouvons là l'expression de la rotation introduite dans la section 3.2.1 concernant les rotations (avec l'équation (3.14)). Nous avons donc :

$$[\boldsymbol{\omega}]_\times = \dot{\mathbf{R}}\mathbf{R}^\top. \qquad (3.41)$$

Cas particulier de la rotation à 1 degré de liberté Supposons par exemple une articulation avec un unique d.d.l. selon l'axe $\boldsymbol{k}$. Alors,

$$[\boldsymbol{\omega}]_\times = \dot{\Theta}\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (3.42)$$

Nous utiliserons cet exemple pour la modélisation de la chaîne cinématique.

3.4.2 La vitesse de déplacement rigide

Nous considérons maintenant que les référentiels $\mathcal{F}$ et $\mathcal{M}$ n'ont plus d'origine commune. Nous avons vu que :

$$\boldsymbol{X}^{\mathcal{F}} = \mathbf{R}\boldsymbol{X}^{\mathcal{M}} + \boldsymbol{t}. \qquad (3.43)$$

La vitesse de déplacement du point X est donc :

$$\dot{\boldsymbol{X}}^{\mathcal{F}} = \dot{\mathbf{R}}\boldsymbol{X}^{\mathcal{M}} + \dot{\boldsymbol{t}}, \qquad (3.44)$$

car $\boldsymbol{X}^{\mathcal{M}}$ est fixe dans le repère $\mathcal{M}$. De même que pour la vitesse de rotation, et en considérant l'équation (3.25), la vitesse de déplacement d'un objet rigide peut s'écrire sous la forme :

$$\begin{aligned} \dot{\boldsymbol{X}}^{\mathcal{F}} &= \widehat{\mathbf{R}}(\boldsymbol{X}^{\mathcal{F}} - \boldsymbol{t}) + \dot{\boldsymbol{t}} \\ &= \boldsymbol{\omega_s} \times (\boldsymbol{X}^{\mathcal{F}} - \boldsymbol{t}) + \dot{\boldsymbol{t}} \\ &= \begin{bmatrix} [\boldsymbol{\omega}]_\times & \dot{\mathbf{t}} \\ \mathbf{0}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}^{\mathcal{F}} - \boldsymbol{t} \\ 1 \end{bmatrix} \\ &= \widehat{\mathbf{D}} \begin{bmatrix} \boldsymbol{X}^{\mathcal{F}} - \boldsymbol{t} \\ 1 \end{bmatrix}. \end{aligned} \tag{3.45}$$

$\widehat{\mathbf{D}} = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \dot{\mathbf{t}} \\ \mathbf{0}^\top & 0 \end{bmatrix}$ est l'**opérateur tangent** du mouvement rigide.

En écrivant

$$\dot{\boldsymbol{t}} = \boldsymbol{v} = v_i \boldsymbol{e}_i + v_j \boldsymbol{e}_j + v_k \boldsymbol{e}_k, \tag{3.46}$$

et en utilisant l'identité $[\boldsymbol{a}]_\times \boldsymbol{b} = -[\boldsymbol{b}]_\times \boldsymbol{a}$ nous pouvons réécrire l'équation (3.45) sous la forme :

$$\boxed{\dot{\boldsymbol{X}}^{\mathcal{F}} = \begin{bmatrix} [\boldsymbol{t} - \boldsymbol{X}^{\mathcal{F}}]_\times & \mathbf{I}_{3\times 3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}}. \tag{3.47}$$

La vitesse de déplacement d'un point rigidement attaché à un objet est donc fonction de la vitesse angulaire de l'objet, de la vitesse de translation de l'objet, mais aussi de la position du point.

Remarque importante : Le vecteur $(\boldsymbol{\omega}, \boldsymbol{v})^\top$ peut être écrit de la manière suivante :

$$\begin{bmatrix} \boldsymbol{\omega}_s \\ \boldsymbol{v}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_i & \boldsymbol{\omega}_j & \boldsymbol{\omega}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_i & \boldsymbol{e}_j & \boldsymbol{e}_k \end{bmatrix} \begin{bmatrix} \dot{\Phi} \\ \dot{\Psi} \\ \dot{\Theta} \\ v_i \\ v_j \\ v_k \end{bmatrix} \tag{3.48}$$

$$= \mathbf{J}_{rigide} \dot{\mathbf{\Gamma}}, \tag{3.49}$$

où $\mathbf{J}_{rigide}$ est la Jacobienne reliant la variation des paramètres de pose ($\dot{\mathbf{\Gamma}}$) au torseur cinématique de l'objet $(\boldsymbol{\omega}, \boldsymbol{v})^\top$, que nous pouvons expliciter :

$$[\boldsymbol{\omega}_i]_\times = [\boldsymbol{e}_i]_\times, \tag{3.50}$$

$$[\boldsymbol{\omega}_j]_\times = \mathbf{R}_i(\Psi)[\boldsymbol{e}_j]_\times \mathbf{R}_i^{-1}(\Psi), \tag{3.51}$$

$$[\boldsymbol{\omega}_k]_\times = \mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)[\boldsymbol{e}_k]_\times \mathbf{R}_j^{-1}(\Phi)\mathbf{R}_i^{-1}(\Psi). \tag{3.52}$$

Chaque colonne de la matrice $\mathbf{J}_{rigide}$ est le *twist* associé au paramètre.

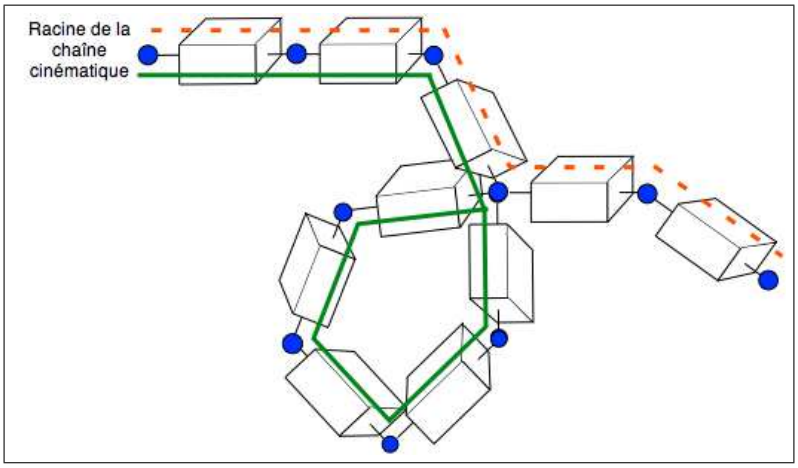


FIG. 3.1: La chaîne cinématique dessinée ici est composée de deux sous-chaînes, l'une étant ouverte (chemin en pointillé), l'autre étant fermée (chemin en gras).

3.5 Modélisation du mouvement articulé

Nous avons explicité la modélisation du mouvement rigide d'un objet. Nous allons maintenant aborder celle du mouvement articulé. Un mouvement articulé est composé de plusieurs mouvements rigides liés entre eux selon une *chaîne* ou un *graphe* cinématique. Nous n'étudierons ici en détails que la modélisation de chaînes cinématiques ouvertes. Par opposition aux chaînes cinématique fermées, elles ne contiennent pas de cycles (c.f. figure 3.1).

De manière générale, une chaîne cinématique est décrite par la taille des éléments la composant, le nombre d'articulations qu'elle comprend et enfin le nombre de d.d.l. de chaque articulation. Si nous parcourons la chaîne cinématique de la racine vers une articulation considérée, nous désignerons par *articulation mère* l'articulation précédente et *articulation fille* l'articulation suivante. La dernière articulation de la chaîne sera nommée *articulation extrémale*. Enfin, l'axe principal d'une articulation sera l'axe joignant l'articulation à son articulation fille (ou l'extrémité de la chaîne dans le cas de l'articulation extrémale).

Notons $\mathbf{\Phi}$ le vecteur des paramètres de la chaîne articulaire. Parmi, ces paramètres, nous distinguons les paramètres de pose que nous notons $\mathbf{\Lambda} = (\lambda_0, \lambda_1, \ldots, \lambda_m)$ et les paramètres de mouvement libre de la racine de la chaîne que nous notons $\mathbf{\Gamma} = (\gamma_0, \ldots, \gamma_q)$. Nous avons donc $\mathbf{\Phi} = (\mathbf{\Lambda}, \mathbf{\Gamma})$. De manière générale, le mouvement libre est composé de trois d.d.l. en translation et de trois d.d.l. en rotation et donc $q = 6$. m est le nombre de degrés de rotation de la chaîne cinématique. Nous noterons $p = m + q$ le nombre de d.d.l. de la chaîne articulaire ayant N éléments (généralement $N \neq p$). Enfin, chaque élément de la chaîne est muni d'un repère (noté $\mathcal{A}_l$ pour l'articulation l) dont l'origine est située sur l'articulation associée.

Nous évoquerons dans un premier temps le choix des repères initiaux pour expri-

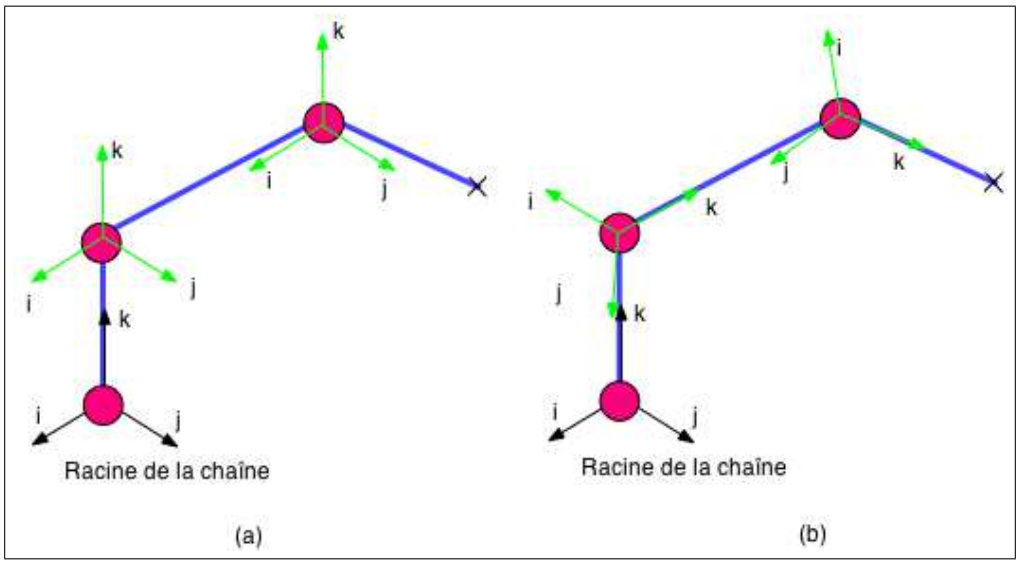


FIG. 3.2: Convention d'orientation des repères pour la chaîne cinématique. (a) En position de référence, tous les repères sont alignés. (b) L'axe $\boldsymbol{k}$ est orienté vers l'articulation fille.

mer les positions et les rotations d'une chaîne cinématique. Puis nous aborderons la modélisation d'une chaîne cinématique. Enfin, nous établirons la Jacobienne de la chaîne cinématique en s'aidant de la modélisation en référence zéro. La Jacobienne permet de faire le lien entre la vitesse de déplacement d'un point la chaîne articulaire et les paramètres de pose de la chaîne.

3.5.1 Choix des repères initiaux

Le choix des repères initiaux (lorsque la chaîne cinématique est dans la position de référence) dans une chaîne articulée est déterminante pour la manipulation des différents éléments la composant. Deux conventions semblent être les plus souvent utilisées :

1. Lorsque la chaîne cinématique est dans sa position initiale, tous les repères sont obtenus par une translation du repère associé à la racine (c.f. figure 3.2-a). Ce choix est avantageux pour effectuer des opérations de cinématique inverse. Cette convention est notamment utilisée dans les logiciels d'animation (*3D* Studio Max ([1]), Maya ([100]), MotionBuilder ([108])), mais aussi dans la norme H-ANIM ([67]).
2. Lorsque la chaîne cinématique est dans sa position initiale, un des axes du repère associé à l'articulation est aligné avec l'axe principal correspondant (c.f. figure 3.2-b). Cette convention est très utilisée en robotique ([109]).

La première convention permet, par exemple, de faire en sorte que toutes les valeurs de $\boldsymbol{\Lambda}$ soient nulles pour la pose initiale. Cependant, la manipulation des mouvements

n'est pas simple. En effet, pour effectuer une rotation autour de l'axe principal de l'articulation, toutes les variables articulaires (de l'articulation) entrent en jeu. Dans le cadre des logiciels d'animation, cette complexité n'est pas visible pour l'utilisateur, puisque généralement ce dernier déplace des objets et le logiciel calcule les transformations induites sur la chaîne cinématique par cinématique inverse.

La seconde convention permet de manipuler plus facilement les différents d.d.l. Par exemple, la rotation autour de l'axe principal de l'élément se fait en modifiant uniquement la valeur d'un seul angle (par exemple si $\boldsymbol{k}$ est aligné avec l'axe principal, alors seul Θ est à modifier). D'autre part, les contraintes angulaires sont simples à mettre en place et à imposer lors de l'estimation des différents paramètres de pose.

Pour représenter notre chaîne articulaire, nous avons décidé d'utiliser la seconde convention. Nous alignons l'axe $\boldsymbol{k}$ avec l'axe principal de l'articulation. Nous verrons que ce choix facilitera les calculs notamment dans le chapitre 4.

Maintenant que nous avons décrit les conventions d'orientation utilisées, nous allons aborder la description mathématique d'une chaîne articulaire.

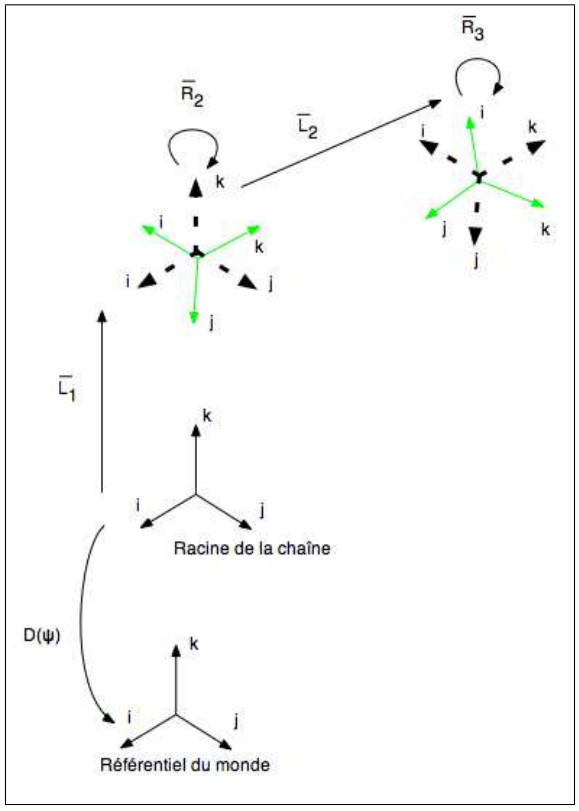


FIG. 3.3: Représentation d'une chaîne cinématique avec deux articulations et un mouvement libre. Les repères en trait plein sont les repères réels de la chaîne tandis que ceux en pointillés permettent d'illustrer les translations induites par les matrices $\overline{\mathbf{L}}_i$.

3.5.2 Coordonnées relatives et absolues d'un point de la chaîne articulaire

Considérons un point X dont les coordonnées sont exprimées dans le repère associé à l'articulation extrémale $(\mathcal{A}_N)$. Les coordonnées de ce point dans le repère de la racine $(\mathcal{R})$ sont :

$$\overline{\boldsymbol{X}}^{\mathcal{R}} = \overline{\mathbf{L}}_1\overline{\mathbf{R}}_2\ldots\overline{\mathbf{L}}_{l-1}\overline{\mathbf{R}}_l\ldots\overline{\mathbf{L}}_{m-1}\overline{\mathbf{R}}_m\overline{\boldsymbol{X}}^{\mathcal{A}_N} \tag{3.53}$$

$$= \mathbf{K}(\boldsymbol{\Lambda})\overline{\boldsymbol{X}}^{\mathcal{A}_N}, \tag{3.54}$$

où :

– $\overline{\mathbf{L}}_l$ est une translation rigide et fixe, de la forme :

$$\overline{\mathbf{L}}_l = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_l \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.55}$$

et l_l est la distance entre l'articulation l et son articulation fille (c.f. figure 3.3),

– $\overline{\mathbf{R}}_l$ est une rotation pure, de la forme (si la rotation a trois d.d.l.) :

$$\overline{\mathbf{R}}_l = \begin{bmatrix} \mathbf{R}_i(\lambda_l)\mathbf{R}_j(\lambda_{l+1})\mathbf{R}_k(\lambda_{l+2}) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{3.56}$$

$\overline{\boldsymbol{X}}^{\mathcal{R}}$ est appelé **coordonnées relatives** de X par rapport à la racine de la chaîne. Bien sûr, les coordonnées de X peuvent être exprimées dans n'importe quel autre repère associé à la chaîne cinématique.

Dans le repère du monde (noté $\mathcal{W}$), nous avons :

$$\overline{\boldsymbol{X}}^{\mathcal{W}} = \mathbf{D}(\boldsymbol{\Gamma})\overline{\boldsymbol{X}}^{\mathcal{R}}, \tag{3.57}$$

où $\mathbf{D}(\boldsymbol{\Gamma})$ est le déplacement rigide de la racine dans le repère du monde et est de la forme :

$$\mathbf{D} = \begin{bmatrix} \mathbf{R}_i(\gamma_3)\mathbf{R}_j(\gamma_4)\mathbf{R}_k(\gamma_5) & \boldsymbol{t}(\gamma_0,\gamma_1,\gamma_2) \\ \mathbf{0}^\top & 1 \end{bmatrix}. \tag{3.58}$$

$\overline{\boldsymbol{X}}^{\mathcal{W}}$ est appelé **coordonnées absolues** du point X. Ce n'est rien d'autre que les coordonnées du point X dans le repère de référence.

De manière générale, pour un point X donné sur une articulation l donnée, nous avons donc :

$$\boxed{\overline{\boldsymbol{X}}^{\mathcal{W}} = \mathbf{D}(\boldsymbol{\Gamma})\mathbf{K}(\boldsymbol{\Lambda}_l)\overline{\boldsymbol{X}}^{\mathcal{A}_l}}, \tag{3.59}$$

où $\boldsymbol{\Lambda}_l$ est le vecteur décrivant les paramètres de pose de l'articulation l.

3.5.3 Modélisation cinématique en référence

Comme nous l'avons vu pour un objet rigide, il est plus compact de représenter la chaîne cinématique en utilisant une modélisation en référence. Nous allons donc développer ici le formalisme de cette modélisation et utiliser celui-ci pour déterminer la Jacobienne de la chaîne (c.f. paragraphe 3.6). Cette Jacobienne nous permettra de faire le lien entre la variation des coordonnées d'un point (dans un référentiel donné) et la variation des paramètres articulaires de la chaîne.

Considérons une position de référence $\overline{\boldsymbol{X}}_0^{\mathcal{W}}$ d'un point X. Cette position est donnée par le vecteur des paramètres de pose $(\boldsymbol{\Gamma}^0, \boldsymbol{\Lambda}^0)$. Nous avons donc : $\overline{\boldsymbol{X}}_0^{\mathcal{W}} = \mathbf{D}(\boldsymbol{\Gamma}^0)\mathbf{K}(\boldsymbol{\Lambda}^0)\overline{\boldsymbol{X}}^{\mathcal{A}_e}$, où $\mathcal{A}_e$ est le repère associé à une articulation extrêmale. Nous pouvons alors reformuler l'équation (3.59) de la manière suivante :

$$\begin{aligned} \overline{\boldsymbol{X}}^{\mathcal{W}}(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}) &= \mathbf{D}(\boldsymbol{\Gamma})\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda^0})\mathbf{D}^{-1}(\boldsymbol{\Gamma^0})\overline{\boldsymbol{X}}_0^{\mathcal{W}} \\ &= \mathbf{H}(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}^0, \boldsymbol{\Lambda}^0)\overline{\boldsymbol{X}}_0^{\mathcal{W}}, \end{aligned} \tag{3.60}$$

$\mathbf{H}$ représente alors la modélisation de la chaîne cinématique en référence. C'est-à-dire que la position d'un point de la chaîne à un instant donné est donné par une transformation des coordonnées de ce point exprimées à un instant antérieur. En reformulant $\mathbf{H}$, nous avons :

$$\begin{aligned} \mathbf{H}(\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{\Gamma}^0, \boldsymbol{\Lambda}^0) &= \mathbf{D}(\boldsymbol{\Gamma}) \underbrace{\mathbf{D}^{-1}(\boldsymbol{\Gamma^0})\mathbf{D}(\boldsymbol{\Gamma^0})}_{\mathbf{I}} \mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda^0})\mathbf{D}^{-1}(\boldsymbol{\Gamma^0}) \\ &= \mathbf{F}(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}^0)\mathbf{Q}(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0, \boldsymbol{\Gamma}^0), \end{aligned} \tag{3.61}$$

avec $\mathbf{F}(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}^0) = \mathbf{D}(\boldsymbol{\Gamma})\mathbf{D}^{-1}(\boldsymbol{\Gamma^0})$ et $\mathbf{Q}(\boldsymbol{\Lambda}, \boldsymbol{\Lambda}^0, \boldsymbol{\Gamma}^0) = \mathbf{D}(\boldsymbol{\Gamma^0})\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda^0})\mathbf{D}^{-1}(\boldsymbol{\Gamma^0})$. Cette reformulation nous permet de séparer le mouvement rigide du mouvement articulaire de la chaîne cinématique. Seul $\mathbf{F}$ dépend du mouvement rigide au cours du temps. $\mathbf{Q}$ ne dépend que des paramètres de pose initiale et des paramètres de la chaîne articulaire. $\mathbf{Q}$ représente donc le mouvement articulaire $(\mathbf{K}(\boldsymbol{\Lambda})\mathbf{K}^{-1}(\boldsymbol{\Lambda^0}))$ exprimé dans un référentiel différent. Nous allons développer $\mathbf{Q}$ et montrer que cette matrice peut s'écrire sous la forme d'un produit de matrices de rotation à un d.d.l.

Analyse de $\mathbf{Q}$

Pour simplifier les notations, notons $\mathbf{J}(\lambda_l) = \overline{\mathbf{R}}_k(\lambda_l)$ la matrice de rotation autour de l'axe $\boldsymbol{k}$. Toute matrice de rotation peut s'écrire comme le produit de matrices de permutation et de matrices de rotation autour de l'axe $\boldsymbol{k}$. Par exemple, une matrice de rotation à trois d.d.l., en convention kji peut être écrite sous la forme :

$$\begin{aligned} \overline{\mathbf{R}}(\lambda_l, \lambda_{l+1}, \lambda_{l+2}) &= \overline{\mathbf{R}}_i(\lambda_l)\overline{\mathbf{R}}_j(\lambda_{l+1})\overline{\mathbf{R}}_k(\lambda_{l+2}) && (3.62) \\ &= \mathbf{TJ}(\lambda_l)\mathbf{TJ}(\lambda_{l+1})\mathbf{TJ}(\lambda_{l+2}), && (3.63) \end{aligned}$$

avec $\mathbf{T}$ de la forme :

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.64}$$

Un autre exemple est celui de la matrice de rotation à un d.d.l. selon l'axe $\boldsymbol{i}$:

$$\overline{\mathbf{R}}(\lambda_l) = \mathbf{TJ}(\lambda_l)\mathbf{TT}, \tag{3.65}$$

avec $\mathbf{T}$ de la même forme que précédemment.

Nous pouvons alors reformuler $\mathbf{K}(\mathbf{\Lambda})$ en utilisant cette composition de matrices à un seul d.d.l. :

$$\mathbf{K}(\mathbf{\Lambda}) = \overline{\mathbf{L}}_1 \underbrace{\mathbf{TJ}(\lambda_0)\mathbf{TJ}(\lambda_1)\mathbf{TJ}(\lambda_2)}_{\text{3 d.d.l.}} \overline{\mathbf{L}}_2 \ldots \overline{\mathbf{L}}_{l-1} \underbrace{\mathbf{TJ}(\lambda_i)\mathbf{TT}}_{\text{1 d.d.l.}} \overline{\mathbf{L}}_l \ldots . \tag{3.66}$$

Aussi, pour la modélisation en référence, nous obtenons une expression de la forme :

$$\begin{aligned} \mathbf{K}(\mathbf{\Lambda})\mathbf{K}^{-1}(\mathbf{\Lambda^0}) \;=\; & \overline{\mathbf{L}}_1\mathbf{TJ}(\lambda_0)\mathbf{TJ}(\lambda_1)\mathbf{TJ}(\lambda_2)\overline{\mathbf{L}}_2 \ldots \\ & \overline{\mathbf{L}}_{N-1}\mathbf{TJ}(\lambda_m)\mathbf{TJ}(\lambda_{m+1})\mathbf{TJ}(\lambda_{m+2}) \\ & \mathbf{J}^{-1}(\lambda^0_{m+2})\mathbf{T}^{-1}\mathbf{J}^{-1}(\lambda^0_{m+1})\mathbf{T}^{-1}\mathbf{J}^{-1}(\lambda^0_m)\mathbf{T}^{-1}\overline{\mathbf{L}}^{-1}_{N-1} \ldots \\ & \overline{\mathbf{L}}^{-1}_2\mathbf{J}^{-1}(\lambda^0_2)\mathbf{T}^{-1}\mathbf{J}^{-1}(\lambda^0_1)\mathbf{T}^{-1}\mathbf{J}^{-1}(\lambda^0_0)\mathbf{T}^{-1}\overline{\mathbf{L}}^{-1}_1, \end{aligned} \tag{3.67}$$

quitte à ce que certaines matrices $\mathbf{J}(\lambda_i)$ soient la matrice identité.

Cette formulation peut être modifiée pour être mise sous la forme d'un produit de matrices à un d.d.l. Nous pouvons dans un premier temps constater que $\mathbf{J}^{-1}(\lambda) = \mathbf{J}(-\lambda)$. Prenons maintenant une articulation i, ayant λ_l, λ_{l+1}, λ_{l+2} comme d.d.l. en rotation. Nous pouvons écrire :

$$\begin{aligned} \overline{\mathbf{L}}_{i-1}\mathbf{TJ}(\lambda_l)\mathbf{TJ}(\lambda_{l+1})\mathbf{TJ}(\lambda_{l+2}) \;=\; & \mathbf{U}_l\mathbf{J}(\lambda_l - \lambda^0_l)\mathbf{U}^{-1}_l\mathbf{U}_{l+1}\mathbf{J}(\lambda_{l+1} - \lambda^0_{l+1})\mathbf{U}^{-1}_{l+1} \\ & \mathbf{U}_{l+2}\mathbf{J}(\lambda_{l+2} - \lambda^0_{l+2})\mathbf{U}^{-1}_{l+2}, \end{aligned} \tag{3.68}$$

avec $\mathbf{U}_l$ de la forme :

$$\mathbf{U}_l \;=\; \mathbf{D}(\mathbf{\Gamma}^0)\mathbf{TJ}(\lambda^0_0)\ldots\mathbf{TJ}(\lambda^0_{l-1})\overline{\mathbf{L}}_{i-1}\mathbf{T}. \tag{3.69}$$

Nous pouvons donc écrire $\mathbf{Q}$ comme le produit de matrices de rotation à un d.d.l. :

$$\mathbf{Q}(\mathbf{\Lambda}, \mathbf{\Lambda}^0, \mathbf{\Gamma}^0) = \mathbf{Q}_1(\lambda_1 - \lambda^0_1, \mathbf{\Gamma^0})\ldots\mathbf{Q}_l(\lambda_l - \lambda^0_l, \mathbf{\Gamma^0})\ldots\mathbf{Q}_m(\lambda_m - \lambda^0_m, \mathbf{\Gamma^0}), \tag{3.70}$$

avec $\mathbf{Q}_l(\lambda_l - \lambda^0_l, \mathbf{\Gamma}^0)$ de la forme :

$$\mathbf{Q}_l(\lambda_l - \lambda^0_l, \mathbf{\Gamma}^0) = \mathbf{U}_l\mathbf{J}(\lambda_l - \lambda^0_l)\mathbf{U}^{-1}_l. \tag{3.71}$$

$\mathbf{Q}_l(\lambda_l - \lambda_l^0, \mathbf{\Gamma}^0)$ est le conjugué de $\mathbf{J}(\lambda_l - \lambda_l^0)$. $\mathbf{U}_l$ ne dépend que des paramètres de pose initiaux et reste donc constant quelque soit le mouvement imprimé à la chaîne articulée.

Enfin, si nous prenons $\mathbf{\Lambda}^0 = \mathbf{0}$, alors nous avons :

$$\mathbf{Q}(\mathbf{\Lambda}, \mathbf{\Gamma}^0) = \mathbf{Q}_1(\lambda_1, \mathbf{\Gamma}^0) \dots \mathbf{Q}_l(\lambda_l, \mathbf{\Gamma}^0) \dots \mathbf{Q}_m(\lambda_m, \mathbf{\Gamma}^0), \tag{3.72}$$

et

$$\boxed{\mathbf{H}(\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{\Gamma}^0) = \mathbf{F}(\mathbf{\Gamma}, \mathbf{\Gamma}^0)\mathbf{Q}_1(\lambda_1, \mathbf{\Gamma}^0) \dots \mathbf{Q}_l(\lambda_l, \mathbf{\Gamma}^0) \dots \mathbf{Q}_p(\lambda_p, \mathbf{\Gamma}^0)}. \tag{3.73}$$

$\mathbf{H}(\mathbf{\Gamma}, \mathbf{\Lambda}, \mathbf{\Gamma}^0)$ est alors la **modélisation de la chaîne cinématique en référence zéro**.

Remarque importante sur l'implémentation : Comme nous l'avons vu en début de chapitre, l'implémentation est effectuée avec les angles d'Euler. Nous pouvons maintenant justifier ce choix par le nombre d'opérations nécessaires pour calculer les différents $\mathbf{U}_i$, ainsi que par la nécessité de décomposer toute la chaîne cinématique en rotations à un d.d.l. Notons aussi que les calculs ne sont pas fait de manière explicite dans le code, mais de manière algorithmique. A aucun moment les matrices pour chacune des articulations ne sont données explicitement. Nous nous contentons de chaîner les transformations (multiplications matricielles) pour effectuer les changements de repère nécessaires.

3.6 Jacobienne de la chaîne cinématique

Nous allons maintenant expliciter la Jacobienne de la chaîne cinématique que nous avons modélisé dans les paragraphes précédents. L'objectif est d'établir explicitement la variation de $\mathbf{H}$ en fonction de la variation des paramètres de pose $\mathbf{\Gamma}$ et $\mathbf{\Lambda}$ de la chaîne articulée. Nous notons $\mathbf{\Phi} = (\mathbf{\Gamma}, \mathbf{\Lambda})$. Il s'agit alors de déterminer la matrice $\mathbf{J}_H$ telle que :

$$d\mathbf{H} = \mathbf{J}_H d\mathbf{\Phi}. \tag{3.74}$$

Soit un point d'un des éléments de la chaîne cinématique. Nous pouvons exprimer la vitesse de déplacement de ce point aussi bien avec six paramètres comme nous l'avons fait dans le paragraphe 3.4.2 (cas de l'objet rigide) qu'en fonction de tous les paramètres de la chaîne articulaire. La Jacobienne tel que nous l'avons défini ci-dessus, permet de faire le lien entre les deux représentations.

Si nous posons $(\boldsymbol{\omega}, \boldsymbol{v})^\top$ le torseur cinématique associé au point, nous avons :

$$\begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{pmatrix} = \mathbf{J}_H \dot{\mathbf{\Phi}}. \tag{3.75}$$

La Jacobienne ne dépend pas du point choisi mais uniquement des paramètres intrinsèques (mécaniques et géométriques ou encore des dimensions et des d.d.l.) de la chaîne articulaire.

De la même manière que pour le cas de l'objet rigide, pour calculer $\mathbf{J}_H$, nous calculons l'opérateur tangent de $\mathbf{H}$, soit $\widehat{\mathbf{H}}$.

Nous avons défini $\mathbf{H} = \mathbf{FQ}$, donc $\dot{\mathbf{H}} = \dot{\mathbf{F}}\mathbf{Q} + \mathbf{F}\dot{\mathbf{Q}}$. Par définition, nous avons $\widehat{\mathbf{H}} = \dot{\mathbf{H}}\mathbf{H}^{-1}$. Nous avons donc :

$$\widehat{\mathbf{H}} = \widehat{\mathbf{F}} + \mathbf{F}\widehat{\mathbf{Q}}\mathbf{F}^{-1}. \tag{3.76}$$

Nous devons donc calculer $\widehat{\mathbf{F}}$ et $\widehat{\mathbf{Q}}$.

Le mouvement libre Nous avons $\widehat{\mathbf{F}} = \dot{\mathbf{D}}(\mathbf{\Gamma})\mathbf{D}^{-1}(\mathbf{\Gamma^0})\mathbf{D}(\mathbf{\Gamma^0})\mathbf{D}^{-1}(\mathbf{\Gamma}) = \widehat{\mathbf{D}}$. Dans le paragraphe 3.4.2, nous avons établi l'expression de $\widehat{\mathbf{D}}$ que nous rappelons ici :

$$\widehat{\mathbf{D}} = \begin{bmatrix} [\boldsymbol{\omega}_r]_\times & \boldsymbol{v}_r \\ \mathbf{0}^\top & 0 \end{bmatrix},$$

où l'indice « $_r$ » dénote le mouvement rigide de la racine. D'autre part, nous avons aussi donné l'expression du torseur cinématique rigide en fonction des paramètres de pose d'une chaîne articulaire :

$$\begin{bmatrix} \boldsymbol{\omega}_r \\ \boldsymbol{v}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\omega}_i & \boldsymbol{\omega}_j & \boldsymbol{\omega}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 \end{bmatrix} \dot{\mathbf{\Gamma}} \tag{3.77}$$

avec :

$$\begin{aligned} [\boldsymbol{\omega}_i]_\times &= [\boldsymbol{e}_i]_\times \\ [\boldsymbol{\omega}_j]_\times &= \mathbf{R}_i(\gamma_4)[\boldsymbol{e}_j]_\times \mathbf{R}_i^{-1}(\gamma_4) \\ [\boldsymbol{\omega}_k]_\times &= \mathbf{R}_i(\gamma_4)\mathbf{R}_j(\gamma_5)[\boldsymbol{e}_k]_\times \mathbf{R}_j^{-1}(\gamma_5)\mathbf{R}_i^{-1}(\gamma_4) \end{aligned}$$

Le mouvement articulaire Nous nous intéressons au calcul de $\widehat{\mathbf{Q}}$. La dérivée de $\mathbf{Q}$ est la dérivée d'un produit de matrices et est donc de la forme :

$$\dot{\mathbf{Q}} = \dot{\mathbf{Q}}_0 \dots \mathbf{Q}_l \dots \mathbf{Q}_m + \dots + \mathbf{Q}_0 \dots \dot{\mathbf{Q}}_l \dots \mathbf{Q}_m + \dots .$$

Par conséquence, $\widehat{\mathbf{Q}}$ est de la forme :

$$\widehat{\mathbf{Q}} = \dot{\mathbf{Q}}\mathbf{Q}^{-1} = \widehat{\mathbf{Q}}_1 + \mathbf{Q}_1\widehat{\mathbf{Q}}_2\mathbf{Q}_1^{-1} + \dots + \mathbf{Q}_1 \dots \widehat{\mathbf{Q}}_m \dots \mathbf{Q}_1^{-1}. \tag{3.78}$$

Or, $\mathbf{Q}_l = \mathbf{U}_l\mathbf{J}(\lambda_l)\mathbf{U}_l^{-1}$ (c.f. équation (3.71)), et donc $\dot{\mathbf{Q}}_l = \mathbf{U}_l\dot{\mathbf{J}}(\lambda_l)\mathbf{U}_l^{-1}$, puisque $\mathbf{U}_l$ est une matrice constante. Nous avons donc $\widehat{\mathbf{Q}}_l = \mathbf{U}_l\widehat{\mathbf{J}}(\lambda_l)\mathbf{U}_l^{-1}$. Pour déterminer $\widehat{\mathbf{Q}}_l$, nous devons maintenant calculer $\widehat{\mathbf{J}}(\lambda_l)$. La dérivée de $\mathbf{J}$ est de la forme :

$$\dot{\mathbf{J}}(\lambda_l) = \dot{\lambda}_l \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\sin\lambda_l & -\cos\lambda_l & 0 \\ 0 & \cos\lambda_l & -\sin\lambda_l & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.79}$$

et donc : $\widehat{\mathbf{J}}(\lambda_l) = \dot{\lambda}_l \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \dot{\lambda}_l\widetilde{\mathbf{J}}.$

$\widehat{\mathbf{Q}}_l$ peut donc être mise sous la forme :

$$\boxed{\widehat{\mathbf{Q}}_l = \dot{\lambda}_l \mathbf{U}_l \widetilde{\mathbf{J}} \mathbf{U}_l^{-1} = \dot{\lambda}_l \widetilde{\mathbf{Q}}_l}. \tag{3.80}$$

Enfin,

$$\boxed{\widehat{\mathbf{Q}}(\mathbf{\Lambda}) = \dot{\lambda}_1 \widetilde{\mathbf{Q}}_1 + \dot{\lambda}_2 \mathbf{Q}_1 \widetilde{\mathbf{Q}}_2 \mathbf{Q}_1^{-1} + \ldots + \dot{\lambda}_m \mathbf{Q}_1 \ldots \widetilde{\mathbf{Q}}_m \ldots \mathbf{Q}_1^{-1}.} \tag{3.81}$$

$\mathbf{H}$ s'écrit donc sous la forme d'une somme de deux termes : l'opérateur tangent du mouvement rigide plus l'opérateur tangent de la chaîne articulaire (lui-même une somme de termes) :

$$\widehat{\mathbf{H}} = \widehat{\mathbf{D}} + \sum_{i=0}^{m} \widehat{\mathbf{H}}_i. \tag{3.82}$$

$\widehat{\mathbf{H}}_i$ dépend de $\mathbf{\Gamma}_0$ et de $(\lambda_0, \ldots, \lambda_i)$ et peut s'écrire sous la forme :

$$\widehat{\mathbf{H}}_i = \dot{\lambda}_l \mathbf{F} \mathbf{Q}_1 \ldots \widetilde{\mathbf{Q}}_i \ldots \mathbf{Q}_1^{-1} \mathbf{F}^{-1} \tag{3.83}$$

$$= \dot{\lambda}_i \begin{bmatrix} [\boldsymbol{\omega}_i]_\times & \boldsymbol{v}_i \\ \mathbf{0}^\top & 0 \end{bmatrix}. \tag{3.84}$$

En combinant les équations (3.82), (3.84) et (3.77), nous obtenons l'expression suivante pour le torseur cinématique du point :

$$\boxed{\begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{pmatrix} = \begin{bmatrix} \boldsymbol{\omega}_i & \boldsymbol{\omega}_j & \boldsymbol{\omega}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_i & \boldsymbol{e}_j & \boldsymbol{e}_k \end{bmatrix} \dot{\mathbf{\Gamma}} + \begin{bmatrix} \boldsymbol{\omega}_1 & \ldots & \boldsymbol{\omega}_m \\ \boldsymbol{v}_1 & \ldots & \boldsymbol{v}_m \end{bmatrix} \dot{\mathbf{\Lambda}}}, \tag{3.85}$$

et donc :

$$\boxed{\mathbf{J}_H = \begin{bmatrix} \boldsymbol{\omega}_i & \boldsymbol{\omega}_j & \boldsymbol{\omega}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\omega}_1 & \ldots & \boldsymbol{\omega}_m \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{e}_i & \boldsymbol{e}_j & \boldsymbol{e}_k & \boldsymbol{v}_1 & \ldots & \boldsymbol{v}_m \end{bmatrix}}. \tag{3.86}$$

Ces résultats ne sont pas nouveaux et peuvent être retrouvés dans l'ouvrage [109] ou encore dans les travaux de Bregler ([18], [19], [20]), Mikic ([103]), *etc.*

Chapitre 4

Modélisation du corps humain

Sommaire

Résumé

Dans ce chapitre, nous présentons le modèle *3D* que nous utilisons pour effectuer la capture du mouvement. Le modèle *3D* est décrit d'une part par sa chaîne cinématique (le squelette) et d'autre part par sa représentation géométrique (la peau et les muscles).

Nous modélisons le corps humain à l'aide de troncs de cônes à base elliptique. Ce choix est orienté par la simplicité de la projection de ces surfaces dans les images ainsi que la bonne approximation pour la modélisation des parties rigides du corps. Les cônes étant des surfaces développables, ils se projettent dans les images sous la forme de segments. L'utilisation de segments permet d'expliciter de manière analytique le mouvement apparent des cônes dans les images. L'utilisation de cônes à base elliptique et non circulaire permet de modéliser correctement la morphologie du corps et notamment des parties comme le torse.

Dans ce chapitre, nous commençons par détailler la modélisation de la chaîne cinématique humaine. Puis nous nous attardons sur la modélisation mathématique des cônes. Nous explicitons le **paramétrage cinématique des contours** observés dans les images. Nous abordons ensuite l'étude du mouvement apparent du contour dans les images aussi bien en fonction des paramètres de pose du modèle que des dimensions du cône. Nous montrons que le mouvement apparent des contours dans les images est la combinaison de deux types de mouvements : le mouvement rigide du cône (étudié au chapitre 3) ainsi que le mouvement relatif de la caméra par rapport au cône. Chacun de ces deux mouvements s'exprime explicitement en fonction des paramètres de pose de la chaîne cinématique. Le mouvement global d'un contour dans l'image est donné sous la forme :

$$\boxed{\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A}+\mathbf{B})\mathbf{J}_H\dot{\boldsymbol{\Phi}}}, \tag{4.1}$$

où $\mathbf{J}_I$ est la matrice Jacobienne image, $\mathbf{A}$ est la Jacobienne du mouvement rigide, $\mathbf{B}$ est la Jacobienne du mouvement de glissement. $\mathbf{J}_H$ est la Jacobienne de la chaîne articulaire (c.f. chapitre 3) et $\boldsymbol{\Phi}$ est le vecteur des paramètres de pose de la chaîne articulaire. Nous donnons un formalisme similaire pour l'étude du mouvement des contours apparents en fonction de la variation des dimensions des cônes. Cette dernière formulation est nécessaire pour adapter le modèle *3D* à la morphologie de l'acteur avant d'effectuer le suivi. **Le paramétrage explicite du mouvement des contours dans les images constitue la première contribution majeure de cette thèse**.

Introduction au chapitre

Pour effectuer la capture du mouvement, plusieurs ingrédients sont nécessaires. Il nous faut des données d'entrée ainsi qu'une idée de ce que nous cherchons à capturer dans les images. Nous avons vu dans la partie introductive de cette thèse, que nous utilisons un modèle *3D* pour modéliser l'acteur que nous voulons capturer. D'autre part, nous avons pris le parti de ne pas utiliser de marqueurs mais des données extraites des images comme les silhouettes ou encore les contours. Nous voulons donc mettre en correspondance les données images avec le modèle *3D*.

Dans ce chapitre nous présentons le modèle *3D* que nous utilisons pour effectuer la capture du mouvement. Plus précisément, le modèle du corps humain comprend deux aspects : la chaîne cinématique qui est une représentation du squelette et la représentation géométrique, qui est une représentation de la peau (ou des vêtements). Nous aborderons successivement ces deux aspects. Nous présenterons donc la modélisation cinématique que nous avons choisi pour modéliser le squelette (paragraphe 4.1). Nous y donnerons, entre autres, les solutions techniques que nous avons apporté au problème du *Gimbal Lock* que nous avons évoqué dans le chapitre précédent. Puis, nous expliciterons les choix que nous avons fait pour la modélisation volumique de l'acteur. Nous donnerons alors le paramétrage analytique du modèle (paragraphe 4.2). Ce dernier nous permettra de modéliser de manière explicite et analytique la projection des surfaces sous la forme de contours dans les images (paragraphe 4.3). Afin d'effectuer le suivi, nous avons besoin de différentier les contours par rapport aux paramètres articulaires. Nous présentons une solution analytique originale à ce problème dans la section 4.4 qui constitue la principale contribution de ce chapitre.

4.1 Modélisation cinématique du corps humain

Le nombre de parties du corps que nous modélisons dans le cadre du suivi du mouvement varie en fonction de la finesse avec laquelle nous voulons modéliser le mouvement. Dans le cadre de la thèse, nous avons choisi de représenter le squelette humain à l'aide de vingt et un segments (c.f. figure 4.3). Ces segments sont reliés entre eux par des articulations pouvant compter de zéro à trois degrés de liberté (d.d.l.). D'autre part, nous pouvons distinguer cinq sous-chaînes dans le squelette : le tronc plus les quatre membres (c.f. figure 4.1). La racine de chacune de ces chaînes est située au niveau du pelvis.

Remarque : Nous pourrions aussi décider que pour un mouvement où l'un des membres reste fixe (dans le repère du monde), la racine des chaînes soit située au niveau de l'articulation immobile. Nous pourrions par exemple prendre un des pieds comme racine dans le cas où l'acteur ne se déplacerait pas. Ou encore, nous pourrions prendre une main dans le cas où celle-ci resterait posée sur un objet immobile pendant le mouvement. Ces autres configurations de chaîne cinématique, bien qu'exotiques, permettraient de faciliter les calculs de la matrice Jacobienne puisque le mouvement libre ne serait plus à prendre en compte.

Dans la suite de ce paragraphe, nous allons décrire les différents d.d.l. des sous-chaînes cinématiques. Nous montrerons ensuite comment nous pouvons éviter les problèmes de singularités (*Gimbal Lock*), introduits au paragraphe 3.2.3, pour les articulations ayant trois d.d.l. Enfin, nous aborderons les contraintes que nous pourrions imposer aux articulations pour rendre la modélisation la plus proche possible des contraintes réelles du corps humain.

4.1.1 Les degrés de liberté

La représentation complète de la chaîne cinématique du corps humain comprend jusqu'à deux cents d.d.l. (en y incluant les articulations des mains, des pieds, de la colonne vertébrale complète, etc.)[67]. Cependant, dans le cadre de nos travaux sur le suivi du mouvement, nous ne pouvons raisonnablement pas estimer l'ensemble de ces d.d.l. Nous nous attachons donc à estimer les paramètres essentiels pour que le mouvement resynthétisé soit proche de l'observation. De plus, nous ne pouvons pas suivre de manière simultanée des mouvements fins (comme ceux des doigts de la main) et le mouvement global du corps. En effet, le suivi des mouvements fins nécessite une résolution très élevée ou une observation rapprochée de ceux-ci. Au contraire, le suivi du mouvement de l'acteur nécessite une vue d'ensemble de ce dernier. A moins d'avoir deux systèmes d'acquisition dédiés, combiner le suivi des deux n'est pas faisable. Dans le cadre de la thèse, nous nous sommes limité au suivi de l'ensemble de l'acteur. La main est donc considérée comme rigide au cours du mouvement. Notons tout de même que les techniques de suivi que nous développons sont transposables au cas du suivi des mains.

D'autre part, certains d.d.l. ne sont pas directement observables et donc difficiles à estimer. La colonne vertébrale est dotée d'une flexibilité difficile à capturer sans système spécifique. Nous avons donc le choix entre réduire le nombre de d.d.l. ou alors modéliser la colonne toute entière à l'aide d'une courbe paramétrée de type *spline*. Nous avons décidé d'effectuer une modélisation intermédiaire. La colonne vertébrale est séparée en trois segments (c.f. figure 4.1). Les d.d.l. de ces segments ne sont pas indépendants. L'articulation de l'abdomen dépend de celle du bassin et du sternum. Cette modélisation permet de rendre compte des liens qui existent entre le mouvement de la base du cou et le mouvement de l'ensemble des vertèbres.

Au final, notre squelette est doté des d.d.l. suivants :
- 6 d.d.l. pour le bassin, qui sont les paramètres du mouvement libre du corps.
- 3 d.d.l. pour chaque hanches, 2 d.d.l. pour chaque genoux, 2 d.d.l. pour chaque cheville, soit 14 d.d.l. pour les jambes.
- 1 d.d.l. pour l'abdomen, 2 d.d.l. pour le sternum, 2 d.d.l. pour le cou, 3 d.d.l. pour la tête soit 8 d.d.l. pour le tronc.
- 1 d.d.l. pour chaque clavicule, 3 d.d.l. pour chaque épaule, 2 d.d.l. pour chaque coude et 2 d.d.l. pour chaque poignet soit 16 d.d.l. pour les bras.

Nous avons donc en tout **quarante quatre d.d.l.** à estimer. L'ensemble de ces d.d.l. est récapitulé sur les figures 4.1 et 4.2.

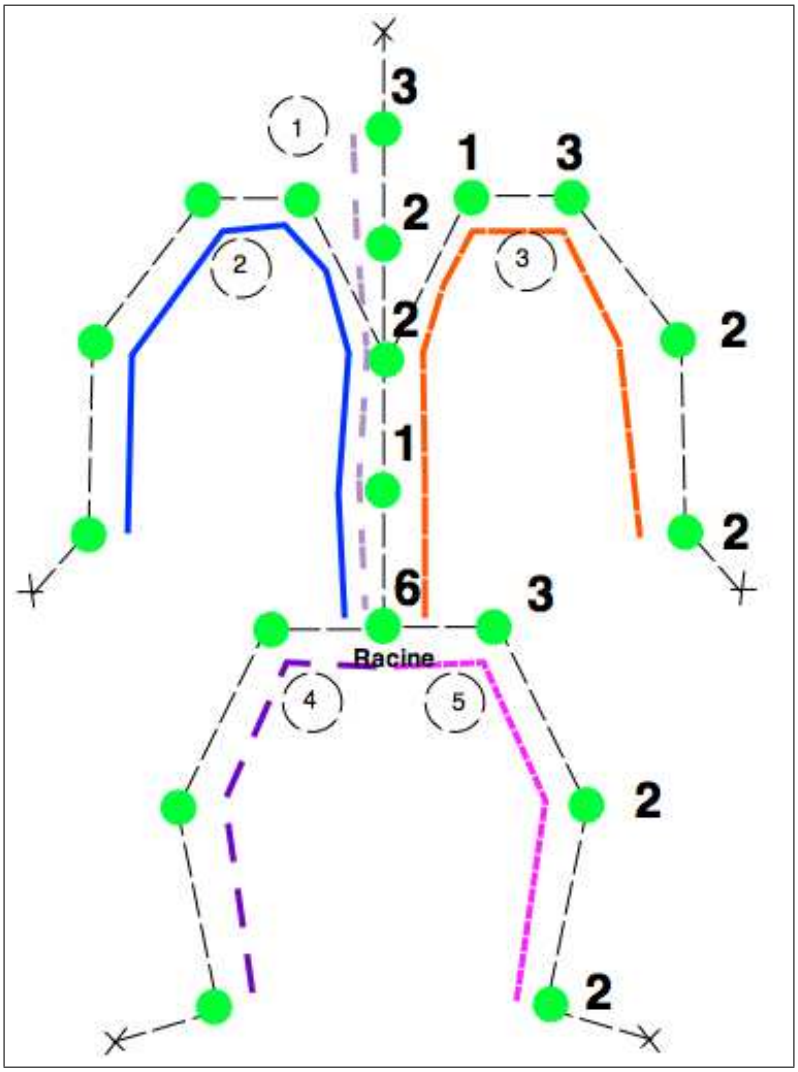


FIG. 4.1: Le corps humain peut être décomposé en 5 chaînes cinématiques. Toutes les chaînes ont une racine commune située au niveau du pelvis. Les chiffres en gras représentent le nombre de degrés de liberté par articulations.

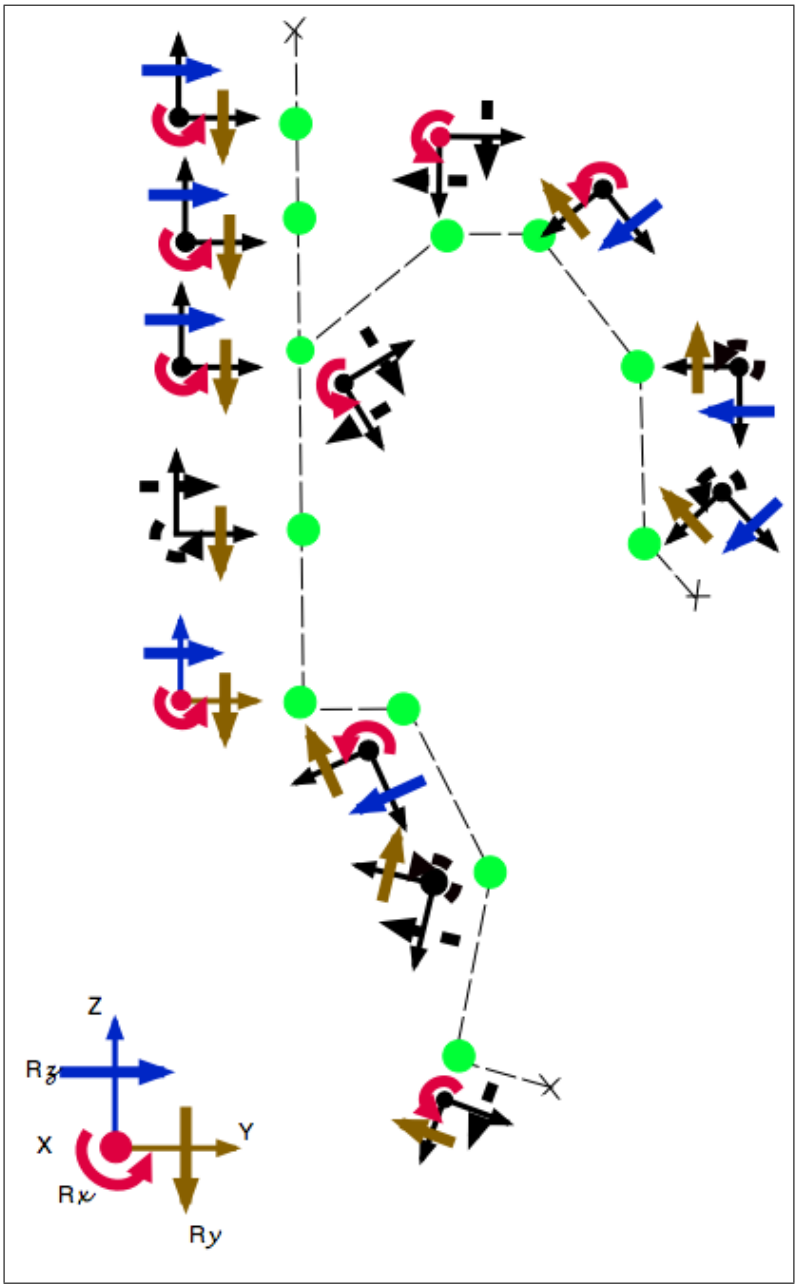


FIG. 4.2: Chaque partie du corps humain est muni d'un repère. Les d.d.l. de chaque articulation sont modélisés par les flèches en couleur, tandis que les flêches en pointillés indiquent une absence de d.d.l.

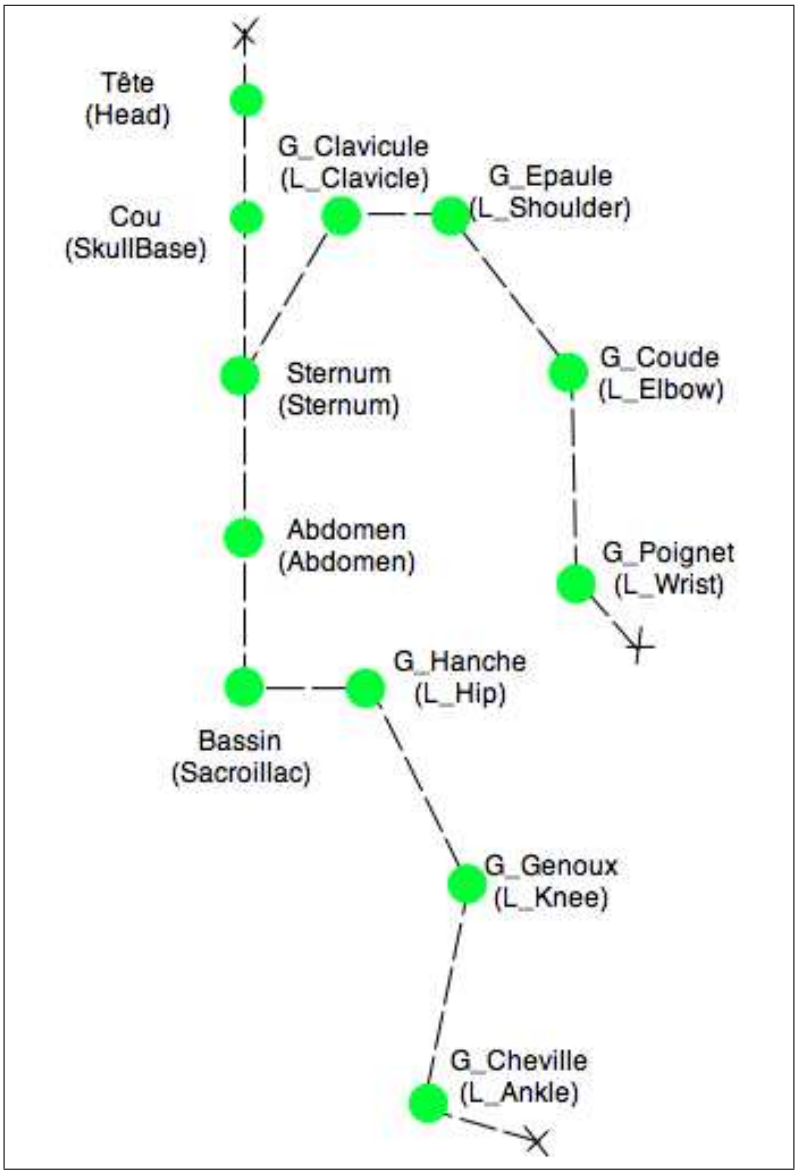


FIG. 4.3: Nous donnons ici les noms des différentes articulations du corps humain. Les membres gauches et droits sont différenciés par un préfixe R_ ou G_.

4.1.2 Le Gimbal Lock

Nous avons vu (chapitre 3 paragraphe 3.2.3) que l'estimation de matrices de rotation avec 3 d.d.l. pouvait entraîner des singularités liés à l'effet de *Gimbal Lock*. Pour remédier à ce problème nous avons mis en place deux stratégies que nous allons maintenant expliciter.

La première consiste à vérifier que les angles estimés n'ont pas des valeurs trop proches des valeurs singulières entraînant un effet de *Gimbal Lock*. Le cas échéant, il s'agit de modifier légèrement les valeurs des angles pour sortir de la singularité. Cette approche nécessite une vérification systématique des valeurs des angles pour chacune des articulations disposant de 3 d.d.l. De plus, si la minimisation fait que la valeur angulaire est entraînée dans cette direction, la contraindre à ne pas y aller peut nuire à l'estimation des paramètres.

L'autre solution consiste à modifier la chaîne cinématique pour répartir les d.d.l. sur plusieurs articulations. Par exemple, pour l'articulation de la hanche, il s'agit de séparer la cuisse en 2 segments, d'enlever la rotation selon l'axe $\boldsymbol{k}$ sur la hanche et de transférer cette rotation sur la seconde partie de la cuisse (c.f. figure 4.4).

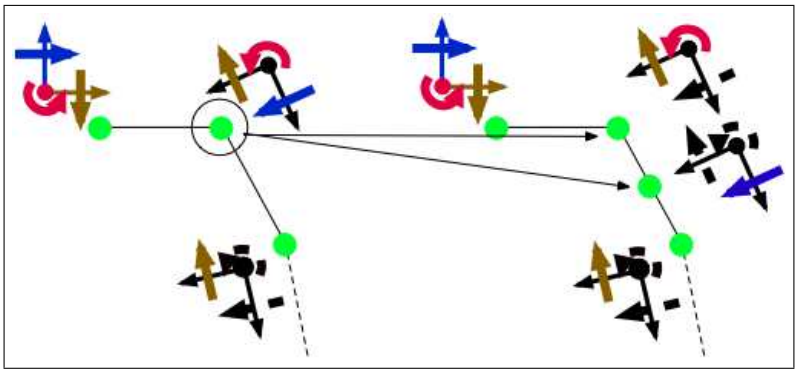


FIG. 4.4: Les articulations munies de 3 d.d.l. sont modifiées et séparées en 2 pour n'avoir que 2 degrés au maximum par articulation. Ici, nous donnons l'exemple de la hanche.

Ces solutions évitent de tomber dans des situations où l'algorithme d'estimation de la pose se retrouve bloqué du fait du manque d'un d.d.l. Par exemple, le pied peut ne pas converger vers une solution correcte du fait que celui-ci soit souvent dans la situation de *Gimbal Lock*. En effet, le pied étant posé sur le sol, il y a un angle de $\pi/2$ entre le pied et le tibia. Cet angle entraîne le *Gimbal Lock* du pied qui n'a plus que 2 d.d.l. La solution consiste à rajouter le troisième d.d.l. sur le tibia.

4.1.3 Contraintes et limites articulaires

Comme nous l'avons vu dans les paragraphes précédents, nous contraignons les chaînes articulées en terme du nombre de d.d.l. Chacune des articulations peut avoir 1,

2 ou 3 d.d.l. Le mouvement humain est aussi limité par des butées articulaires. Avec la représentation angulaire choisie, nous pouvons introduire ces butées articulaires dans les contraintes du mouvement. Ces limites peuvent permettre d'éviter à l'algorithme d'estimer des paramètres articulaires bio-mécaniquement impossibles. Dans l'annexe B.1, nous donnons les différentes valeurs des limites angulaires selon des données recueillies par la NASA.

Cependant, en pratique nous avons décidé de ne pas introduire ces limites dans les contraintes de minimisation. En effet, l'objectif est de suivre un mouvement réalisé par un acteur réel. Le mouvement à suivre respecte donc les contraintes biomécaniques. Par conséquence, si les limitations articulaires ne sont pas respectées après estimation du mouvement, c'est que le suivi du mouvement est erroné. Le mouvement peut cependant paraître correct, mais si nous analysons les valeurs angulaires, nous pouvons constater des effets de *flip* des angles (une rotation de π). Pour remédier à ces effets, nous pouvons bloquer la variation angulaire. Cependant, ce blocage peut être préjudiciable lors de l'estimation des paramètres. Nous nous retrouverions dans le même cas que le *Gimbal Lock*. Nous avons pris le parti de laisser libre les articulations pendant le suivi et de corriger les inversions d'angles à l'aide du logiciel MKM de l'UHB ([91]).

Remarque : Dans la modélisation la plus couramment utilisée, la chaîne cinématique du corps humain n'a pas de chaîne fermée. Cependant, nous pourrions améliorer la modélisation du comportement de l'épaule en créant une contrainte sur la chaîne articulaire permettant de s'approcher d'une chaîne fermée telle qu'illustrée sur la figure 4.5. Celle-ci permet de contraindre le mouvement de l'épaule de manière plus correcte. Afin d'éviter la création d'un cycle nous modélisons cette contrainte par pénalisation.

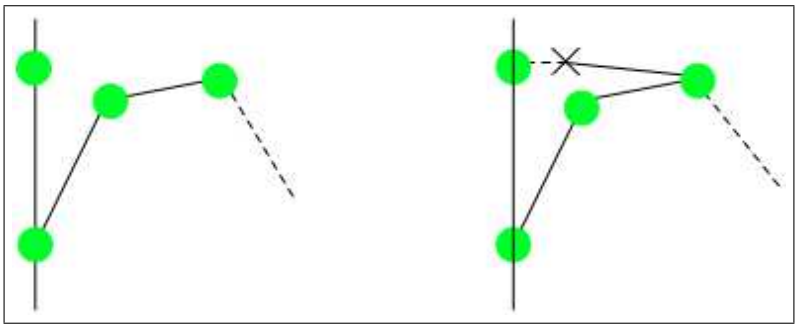


FIG. 4.5: L'épaule peut être modélisée en chaîne ouverte ou en chaîne fermée. Cette seconde modélisation complexifie la représentation, mais modélise mieux les contraintes de mouvement de l'épaule.

4.2 Modélisation géométrique

Dans la partie précédente, nous avons abordé la modélisation de la chaîne cinématique du corps humain. Nous nous sommes attaché à modéliser l'ensemble des

d.d.l. du squelette. Pour effectuer le suivi, nous avons besoin d'une représentation volumique ou surfacique du modèle. Dans le chapitre 2, au paragraphe 2.3.4.2, nous avons abordé une description de différents modèles géométriques utilisés dans les travaux précédents. Nous allons maintenant expliciter et justifier notre choix.

Notre Choix La méthode de suivi de mouvement que nous avons mis en place est une technique utilisant la projection du modèle dans les images. Or une surface *3D* se projette dans les images sous la forme de contours. Nous avons donc besoin de primitives dont la projection dans les images soit assez simple et dont les contours apparent (dans l'image) rendent compte au mieux de la morphologie humaine. Les contours les plus simples étant des droites dans les images, notre choix s'est naturellement porté sur des quadriques dégénérées comme les cylindres ou les cônes. Les cylindres ou cônes à base circulaire n'étant pas adéquats pour modéliser certaines parties du corps comme le torse, nous avons décidé d'utiliser des **cônes tronqués dont la base a une forme elliptique**. La projection de ces cônes dans les images sont des segments de droite, comme nous le verrons dans le paragraphe 4.3.

Nous ne modélisons que les parties visibles du corps humain. Nous avons donc un squelette comportant vingt et un segments mais une modélisation géométrique comportant *dix sept primitives*. A ces dix sept primitives, nous rajoutons le bout des doigt, les chevilles, les épaules et le sommet du crâne, soit un total de vingt quatre primitives pour la modélisation géométrique complète. Pour des raisons de temps de calcul et de complexité, nous utilisons pour des séquences simples le modèle comportant 17 primitives. Le modèle complet a été élaboré avec Loïc Lefort et Franck Multon, sur des bases de modèles approchant au mieux les contraintes anatomiques. Enfin, les primitives ne sont pas jointives sur le modèle *3D* (c.f. figure 4.6). La modélisation au niveau des articulations n'est pas définie dans notre modèle. Ce choix est lié au fait qu'au niveau des articulations, les contours extraits dans les images ne sont pas bien définis. Nous ne modélisons donc l'acteur que pour les parties rigides du corps. C'est sur ces parties rigides (torse inclus) que les contours détectés sont les moins ambiguës. Plus précisément, la difficulté au niveau des articulations est par exemple de déterminer à quelle partie du corps appartient le contour détecté dans l'image. En pratique, nous ne constatons pas de dégradation du suivi du fait que les primitives ne soient pas jointives.

4.2.1 Paramétrage des surfaces

Nous avons choisi de modéliser toutes les parties du corps à l'aide de cônes elliptiques, qui sont des surfaces développables. Nous verrons dans la section 4.4 l'importance de ce choix, qui permet d'exprimer le mouvement des contours apparents. Dans certains cas, ce choix n'est pas idéal. Bien que nous ne les ayons pas utilisés, nous présentons en annexe l'extension aux cas des ellipsoïdes.

Nous présentons maintenant le paramétrage que nous utiliserons tout au long de l'exposé pour décrire et utiliser les cônes lors de la capture du mouvement. Le lecteur pourra noter que, tout comme les ellipsoïdes, les cônes sont des surfaces quadriques.

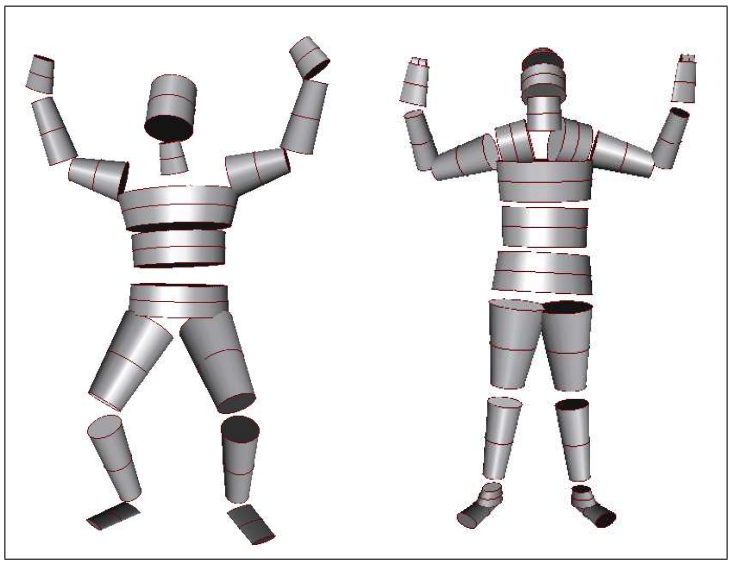


FIG. 4.6: Chacune des parties du corps humain est modélisée par un ou plusieurs cônes elliptiques tronqués. Nous donnons ici la représentation simple et la représentation complète du modèle *3D* utilisé pour effectuer le suivi du mouvement. Sur le modèle complet, nous avons rajouté les épaules, le bout des doigts, les chevilles et le sommet du crâne.

L'extension que nous proposons dans l'annexe B.2 permet de faire ressortir des similarités dans le paramétrage et la projection de ces surfaces dans les images.

Les cônes sont aussi une sous-classe de surfaces dites réglées et plus particulièrement de surfaces développables. Nous allons aborder le paramétrage des cônes vu sous l'angle des surfaces réglées. Nous n'aborderons pas ici le thème du paramétrage de ces surfaces de manière générale, nous ne donnerons que le paramétrage que nous avons utilisé au cours de cette thèse. Pour plus de détails sur le paramétrage des surfaces, le lecteur pourra se référer à [42] ou encore [87].

Définitions

Surface réglée : Une surface réglée est une surface dont tout point X peut être paramétré de la manière suivante :

$$\boldsymbol{X}(u,v) = \boldsymbol{\alpha}(u) + v\boldsymbol{\beta}(u) \tag{4.2}$$

où $\boldsymbol{\alpha}$ est un point *3D* et $\boldsymbol{\beta}$ est un vecteur, et où donc le couple $(\boldsymbol{\alpha}(u), \boldsymbol{\beta}(u))$ décrit une droite paramétrée par u. D'autre part, $\boldsymbol{\alpha}$ et $\boldsymbol{\beta}$ doivent être continus et dérivables par rapport à u.

Courbure Gaussienne : La courbure gaussienne d'une surface peut être définie en utilisant les coefficients de la première et de la seconde forme fondamentale de la

surface :

$$\begin{aligned} l(du,dv) &= d\boldsymbol{X}\cdot d\boldsymbol{X} \\ &= g_{uu}du^2 + g_{uv}ddv + g_{vv}dv^2 \qquad (4.3) \\ ll(du,dv) &= d^2\boldsymbol{X}\cdot\boldsymbol{n} \\ &= L_{uu}d^2u + L_{uv}dudv + L_{vv}d^2v, \qquad (4.4) \end{aligned}$$

où $\boldsymbol{n}$ désigne le vecteur normal au point X et est donné par la formule suivante :

$$\boldsymbol{n} = \frac{\partial \boldsymbol{X}}{\partial v} \times \frac{\partial \boldsymbol{X}}{\partial u} = \boldsymbol{X}_v \times \boldsymbol{X}_u. \qquad (4.5)$$

Ces deux formes fondamentales sont des définitions très utiles et permettent de déduire des métriques sur les surfaces. Pour plus de détails, le lecteur pourra se référer à [42] (pages 92-99 et page 141).
La courbure gaussienne K est donnée par la formule suivante :

$$K = \frac{L_{uu}L_{vv} - L_{uv}}{g_{uu}g_{vv} - g_{uv}}. \qquad (4.6)$$

Si nous considérons le paramétrage de la surface réglée définie avec l'équation (4.2), nous avons :

$$\boldsymbol{n} = (\dot{\boldsymbol{\alpha}}(u) + v\dot{\boldsymbol{\beta}}(u)) \times \boldsymbol{\beta},$$

$$\begin{aligned} g_{uu} &= (\dot{\boldsymbol{\alpha}}(u) + v\dot{\boldsymbol{\beta}}(u))^\top(\dot{\boldsymbol{\alpha}}(u) + v\dot{\boldsymbol{\beta}}(u)), \\ g_{uv} &= 2(\dot{\boldsymbol{\alpha}}(u) + v\dot{\boldsymbol{\beta}}(u))\boldsymbol{\beta}, \\ g_{vv} &= \boldsymbol{\beta}(u)^2, \end{aligned}$$

$$\begin{aligned} L_{uu} &= (\ddot{\boldsymbol{\alpha}}(u) + v\ddot{\boldsymbol{\beta}}(u))^\top(\dot{\boldsymbol{\alpha}}(u) + v\dot{\boldsymbol{\beta}}(u)) \times \boldsymbol{\beta}, \\ L_{uv} &= \det(\begin{bmatrix} \dot{\boldsymbol{\alpha}} & \boldsymbol{\beta} & \dot{\boldsymbol{\beta}} \end{bmatrix})\ (*), \\ L_{vv} &= 0, \end{aligned}$$

où la notation $(\dot{})$ représente la dérivée par rapport à u et $(\ddot{})$ est la dérivée seconde par rapport à u.
$(*)$ est obtenue en considérant les étapes suivantes :

$$\begin{aligned} L_{uv} &= 2\dot{\boldsymbol{\beta}}\cdot(\dot{\boldsymbol{\alpha}}(u) + v\dot{\boldsymbol{\beta}}(u)) \times \boldsymbol{\beta} \\ &= 2\dot{\boldsymbol{\beta}}\cdot\dot{\boldsymbol{\alpha}}(u) \times \boldsymbol{\beta} + \underbrace{2v\dot{\boldsymbol{\beta}}\cdot\dot{\boldsymbol{\beta}}(u) \times \boldsymbol{\beta}}_{0} \\ &= \det(\begin{bmatrix} \dot{\boldsymbol{\alpha}} & \boldsymbol{\beta} & \dot{\boldsymbol{\beta}} \end{bmatrix}) \end{aligned}$$

Surface développable : Une surface développable est une surface réglée dont la courbure gaussienne est nulle. En considérant les expressions précédentes, la condition

$K = 0$ est équivalente à $L_{uv} = 0$ ou encore $\det(\begin{bmatrix} \dot{\boldsymbol{\alpha}} & \boldsymbol{\beta} & \dot{\boldsymbol{\beta}} \end{bmatrix}) = 0$. Cette contrainte est vérifiée si l'un des trois vecteurs est une combinaison linéaire des autres. Nous pouvons alors choisir $\dot{\boldsymbol{\alpha}} = a\boldsymbol{\beta} + b\dot{\boldsymbol{\beta}}$. Nous obtenons ainsi une nouvelle expression pour la normale à la surface :

$$\boldsymbol{n} = (1 + bv)\dot{\boldsymbol{\beta}} \times \boldsymbol{\beta} \tag{4.7}$$

Dans la suite, nous aurons besoin seulement de la direction de la normale $\boldsymbol{n}$. Nous pouvons donc ignorer le facteur d'échelle $(1+bv)$. La direction de la normale à la surface ne dépend que de $\boldsymbol{\beta}$ et $\dot{\boldsymbol{\beta}}$, et donc uniquement du paramètre u. Nous verrons que cette propriété est importante dans la suite de l'exposé.

Application au cône : Le cône est une surface développable. Nous pouvons donc paramétrer le cône de la manière suivante :

$$\boldsymbol{\alpha}(u) = \begin{bmatrix} a\cos(u) \\ b\sin(u) \\ 0 \end{bmatrix} \quad \text{et} \quad \boldsymbol{\beta}(u) = \begin{bmatrix} ak\cos(u) \\ bk\sin(u) \\ 1 \end{bmatrix} . \tag{4.8}$$

a, b sont les demi-axes majeur et mineur du cône, et $k = -\frac{1}{l}$ (c.f. figure 4.7). En général, nous posons $u = \theta$ et $v = z$.

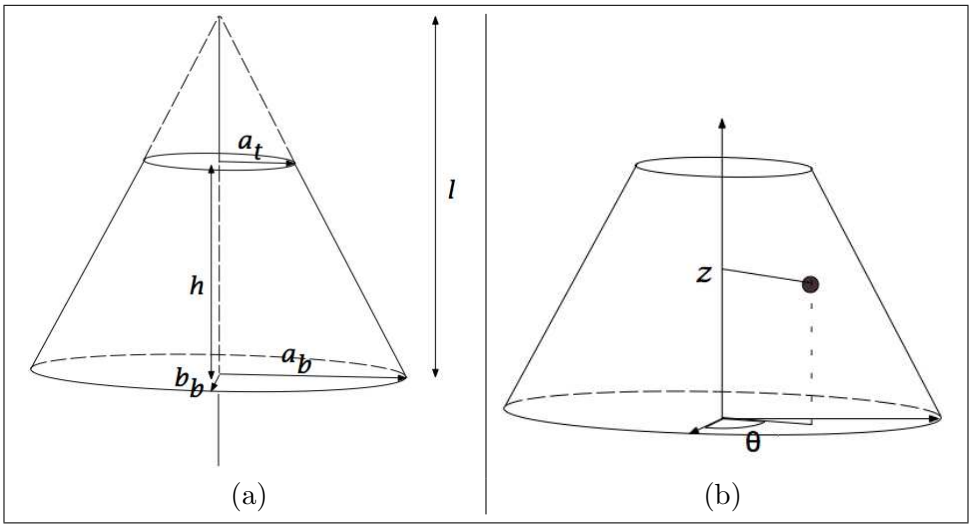


FIG. 4.7: (a) Un cône elliptique tronqué peut être décrit par 4 paramètres : les demi petit et grand axes de la base, la hauteur du cône et le demi grand axe du sommet. (b) Un point de la surface est paramétré par θ et z.

La forme paramétrique d'un cône devient donc :

$$\boldsymbol{X}(\theta, z) = \begin{bmatrix} a(1 + kz)\cos(\theta) \\ b(1 + kz)\sin(\theta) \\ z \end{bmatrix} . \tag{4.9}$$

Après simplification du facteur commun $(1+kz)$, la normale en tout point de la surface du cône a pour coordonnées :

$$\boldsymbol{n}(\theta, z) = \begin{bmatrix} b\cos(\theta) \\ a\sin(\theta) \\ -abk \end{bmatrix}. \tag{4.10}$$

La direction de la normale au cône est donc bien indépendante de la hauteur z.

La forme paramétrique du cône étant relativement simple, nous utiliserons celle-ci dans les développements mathématiques que nous allons aborder maintenant.

4.3 La projection du modèle

Pour effectuer le suivi du mouvement, nous projetons le modèle *3D* dans chacune des images. Nous modélisons le corps humain à l'aide de cônes elliptiques tronqués dont les projections dans les images sont composées de segments de droites et d'ellipses. Dans ce paragraphe, nous nous intéressons au paramétrage des segments en fonction des paramètres de pose du modèle ainsi que des paramètres de calibrage des caméras. Nous justifierons aussi le choix de ne pas considérer les ellipses.

Pour la clarté de l'exposé, nous considérons dans un premier temps un cône projeté dans une image. Nous étendrons au cas du cône appartenant à une chaîne cinématique en fin de section.

4.3.1 Observation du modèle dans les images

Nous allons aborder la modélisation mathématique de la projection d'un cône dans une image. De manière générale, les surfaces se projettent dans les images sous la forme de deux contours :

- les contours de discontinuité,
- les contours extrémaux.

Le premier type de contours est lié à une rupture de continuité sur la surface d'un objet. Cette rupture est liée à l'intersection de deux surfaces. Par exemple le cube produit des contours de discontinuité aux arrêtes. Le cône forme deux contours de discontinuité de forme elliptique à l'intersection entre les surfaces des sommets et la surface principale (c.f. figure 4.8).

Les contours extrémaux sont la projection sur l'image du lieu des points de la surface où le rayon de vue est tangent à celle-ci. Une expression anglaise satisfait mieux l'explication ici : *An extremal contour appears in an image whenever the surface turns smoothly away from the viewer*. Le lieu des points sur la surface où le rayon de vue est tangent à la surface est appelé le « contour occultant » ou *rims*. Plusieurs travaux ont été menés sur le contours.

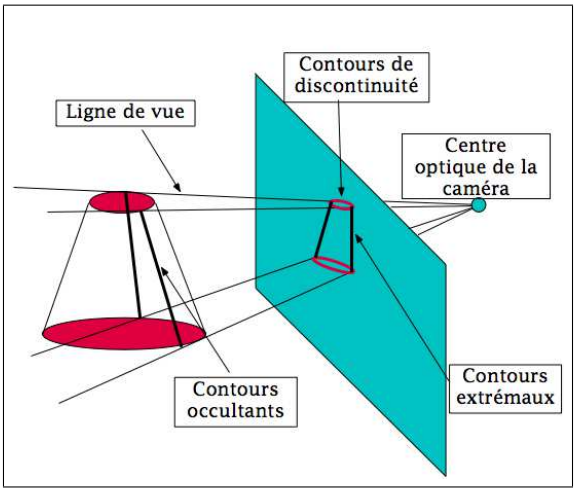


FIG. 4.8: Un cône se projette dans une image sous la forme de deux types de contours : les contours de discontinuité liés à l'intersection de deux surfaces et les contours extrémaux. Ces derniers sont le lieu des points images où la ligne de vue est tangente à la surface.

Nous nous intéresserons ici à ce second type de contours. Les contours de discontinuité pour le cône ne sont en pratique pas observables. En effet, ils sont situés au niveau des articulations et n'ont donc pas de réalité.

Dans un premier temps nous expliciterons la forme analytique des contours occultant. Puis nous établirons l'expression analytique de la projection des points du contour occultant dans les images.

4.3.2 Paramétrage cinématique des contours observés

Considérons un point X sur la surface, de coordonnées $\boldsymbol{X}$ dans le repère du cône. Soit $\mathbf{R}$ la matrice d'orientation de la surface et $\boldsymbol{t}$ la position de celle-ci exprimée par rapport à un référentiel donné (celui du monde par exemple) (c.f. figure 4.9). Alors X appartient au contour occultant s'il vérifie l'équation :

$$(\mathbf{R}\boldsymbol{n})^\top(\mathbf{R}\boldsymbol{X} + \boldsymbol{t} - \boldsymbol{C}) = 0, \tag{4.11}$$

où $\boldsymbol{n}$ est le vecteur normal à la surface au point X et $\boldsymbol{C}$ les coordonnées du centre optique de la caméra exprimées dans le repère du monde. Dans l'équation (4.11), le terme $\mathbf{R}\boldsymbol{X} + \boldsymbol{t} - \boldsymbol{C}$ est le vecteur directeur du rayon de vue exprimé dans le repère du monde et $\mathbf{R}\boldsymbol{n}$ représente les coordonnées du vecteur normal au cône exprimé dans le repère du monde. Nous pouvons aussi formuler l'équation (4.11) de la manière suivante :

$$(\boldsymbol{X} + \mathbf{R}^\top(\boldsymbol{t} - \boldsymbol{C}))^\top\boldsymbol{n} = 0, \tag{4.12}$$

Alors que la première formulation exprime la contrainte dans le référentiel du monde, la seconde formulation exprime la contrainte dans le repère de l'objet.

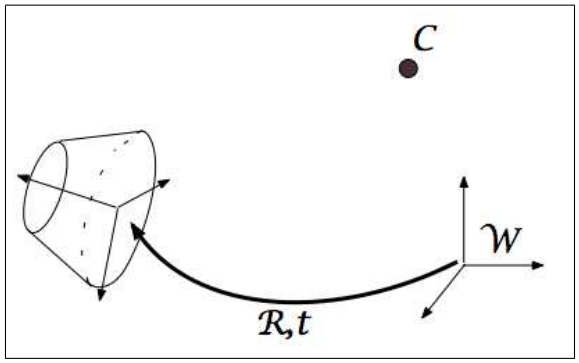


FIG. 4.9: $\mathbf{R}$ et $\boldsymbol{t}$ sont respectivement la rotation et la position du cône exprimées dans le repère du monde $\mathcal{W}$. $\boldsymbol{C}$ est la position du centre optique de la caméra exprimée dans le repère du monde.

La contrainte peut aussi être exprimée en utilisant les formes matricielles des quadriques. Nous n'aborderons pas cette formulation ici. Le lecteur pourra se référer à [29] pour plus de détails.

Pour déterminer l'ensemble des points décrivant les contours occultant de l'objet, nous résolvons l'équation (4.12). Nous aborderons ici le cas du cône elliptique. Pour les ellipsoïdes, le lecteur pourra se référer à l'annexe B.2.

Les contours occultant d'un cône Nous utilisons le paramétrage introduit avec l'équation (4.9). L'équation (4.12) est de la forme :

$$\left(\begin{bmatrix} a(1-\frac{z}{l})\cos(\theta) \\ b(1-\frac{z}{l})\sin(\theta) \\ z \end{bmatrix} + \begin{pmatrix} \boldsymbol{r}_1^\top \\ \boldsymbol{r}_2^\top \\ \boldsymbol{r}_3^\top \end{pmatrix} (\boldsymbol{t}-\boldsymbol{C}) \right)^\top \begin{bmatrix} b\cos(\theta) \\ a\sin(\theta) \\ -abk \end{bmatrix}. \tag{4.13}$$

Si nous développons l'équation (4.13), nous obtenons une équation trigonométrique de la forme :

$$F\sin(\theta) + G\cos(\theta) + H = 0, \tag{4.14}$$

avec :

$$\begin{aligned} F &= b\boldsymbol{r}_1^\top(\boldsymbol{t}-\boldsymbol{C}), \\ G &= a\boldsymbol{r}_2^\top(\boldsymbol{t}-\boldsymbol{C}), \\ H &= -abk\boldsymbol{r}_3^\top(\boldsymbol{t}-\boldsymbol{C}) + ab. \end{aligned} \tag{4.15}$$

L'équation (4.14) ne dépend pas de z. Il s'agit donc d'une équation dont la seule inconnue est θ. La contrainte de tangence est donc indépendante de z. Ainsi quelque soit z, pour un θ solution de cette équation, le point $X(\theta, z)$ appartient au contour occultant. Le lieu des points du contour occultant décrit donc une droite sur la surface du cône. Cette droite passe aussi par le sommet du cône, il s'agit donc d'une génératrice du cône.

Résolution de l'équation (4.14) **:** Pour résoudre cette équation, nous utilisons le changement de variable suivant :

$$t = \tan(\theta/2) \quad , \quad \cos(\theta) = \tfrac{1-t^2}{1+t^2} \quad \text{et} \quad \sin \tfrac{2t}{1-t^2}, \tag{4.16}$$

Ce changement de variable nous permet d'obtenir une équation du second degré en t à résoudre. Cette nouvelle équation est de la forme :

$$(H - F)t^2 + 2Gt + (F + H) = 0. \tag{4.17}$$

Cette équation admet donc zéro, une ou deux solutions. L'étude du nombre de solutions pour cette équation peut se faire de façon géométrique. Pour simplifier le propos, nous allons poser $\mathbf{R} = \mathbf{I}_{3\times3}$ et $\boldsymbol{t} = \mathbf{0}$; en d'autres termes, nous supposons que le repère associé à la surface et le repère du monde sont confondus.

Les coefficients F, G et H deviennent alors :

$$\begin{aligned} F &= -bc_1, \\ G &= -ac_2, \\ H &= ab(1 + kc_3). \end{aligned} \tag{4.18}$$

Le discriminant de l'équation (4.17) est :

$$\begin{aligned} \Delta &= 4G^2 - 4(H^2 - F^2) && (4.19)\\ &= 4(a^2c_2^2 + b^2c_1^2 - a^2b^2(1 + kc_3)^2) && (4.20)\\ &= 4a^2b^2(\frac{c_2^2}{b^2} + \frac{c_1^2}{a^2} - (1 + kc_3)^2). && (4.21) \end{aligned}$$

Pour déterminer le nombre de solutions, il s'agit bien évidemment de regarder le signe de Δ. On peut remarquer que Δ est en fait l'équation du cône considéré avec les coordonnées de la caméra comme inconnues. Le nombre de solutions dépend donc de la position de la caméra par rapport au cône. Si la caméra est à l'intérieur du cône (c.f. figure 4.10 cas n^o 1) il n'y a pas de contours occultant. Si la caméra est sur le cône (c.f. figure 4.10 cas n^o 2) alors $\Delta = 0$: il y a une solution double et donc un seul contour (réduit à un point dans l'image). Enfin, si la caméra est à l'extérieur du cône (c.f. figure 4.10 cas n^o 3) alors $\Delta > 0$ et il y a deux contours occultant.

La position des contours occultant dépend donc de la position de la caméra par rapport au cône. Si nous déplaçons la caméra par rapport au cône, les contours occultant « glissent » le long de la surface. Nous verrons que cet aspect est important dans l'étude du mouvement des contours observés dans les images.

La résolution de l'équation (4.17) étant évidente, nous n'aborderons pas ce point ici. Nous donnons juste la forme de la solution pour information :

$$t_{1,2} = \frac{-2G \pm \sqrt{\Delta}}{2(H - F)}. \tag{4.22}$$

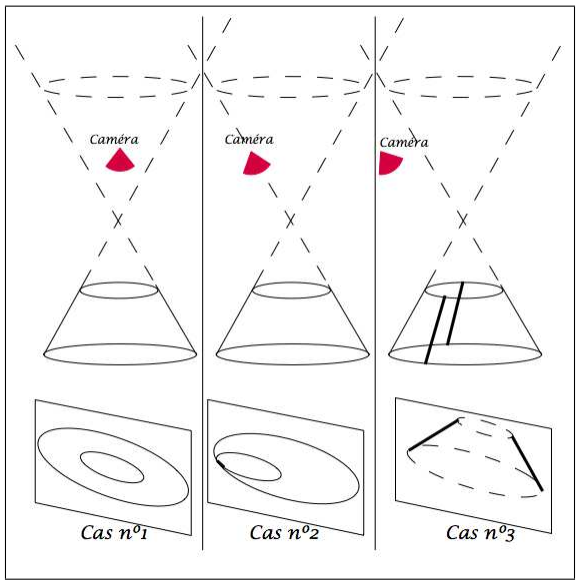


FIG. 4.10: Selon la position de la caméra par rapport au cône, il peut y avoir zéro (cas n° 1), un (cas n° 2) ou deux (cas n° 3) contours occultant. Sur la figure nous illustrons sur la première ligne la position de la caméra en rouge et les contours occultant ; Sur la seconde ligne, la projection de ces contours dans les images.

Or, $t = \tan(\frac{\theta}{2})$ donc $\theta = 2\tan^{-1}(t)$. L'équation (4.17) nous permet d'obtenir un ou deux angles qui décrivent les contours occultant. Nous avons vu que ces contours sont des segments de droite. Pour connaître les coordonnées de tous les points sur le segment, il suffit de connaître les coordonnées de deux d'entre eux (les autres se déduisant par combinaison linéaire). Nous calculons les coordonnées des points pour $z = h$ et pour $z = 0$.

Remarques :
- Si pour le calcul des coordonnées des points la remarque sur la combinaison linéaire n'a pas de grand intérêt, nous verrons que pour déterminer le déplacement des contours cela simplifie grandement les calculs.
- Nous verrons qu'en pratique, il n'est pas nécessaire de connaître θ, il suffit de calculer $\cos(\theta)$ et $\sin(\theta)$ pour déterminer complètement les coordonnées du point X. Ces deux valeurs sont facilement calculables en utilisant les changements de variables introduits dans (4.16).
- Dans la suite de l'exposé nous noterons $\boldsymbol{X}_{occ}(\theta, z)$ les points appartenant aux contours occultant.

Les contours extrémaux d'un cône Maintenant que nous avons une solution analytique pour l'ensemble des points appartenant aux contours occultant, il suffit de les projeter dans les images pour obtenir les contours extrémaux. D'après la remarque faite précédemment, il suffit de calculer la projection des points situés sur la base et au sommet pour déterminer complètement le contour image.

Cependant, la projection n'est pas une opération linéaire. Donc, la projection de points échantillonnés sur le contour occultant n'est pas équivalente à l'échantillonnage des contours extrémaux. Or, nous voulons estimer le mouvement d'un point image en fonction des paramètres du modèle *3D*. Nous avons donc choisi, pour effectuer le suivi, d'échantillonner les contours occultant et de projeter chacun des points du contour dans les images.

Comme nous l'avons dit dans l'introduction de cette thèse, les caméras sont calibrées. En utilisant les notations standards pour représenter la matrice des paramètres intrinsèques de la caméra ($\mathbf{K}$), nous avons :

$$\boldsymbol{x}_{occ} = \mathbf{K}\mathbf{M}\overline{\boldsymbol{X}}_{occ}, \tag{4.23}$$

où $\mathbf{M}$ est une matrice 4×4 associée à la transformation entre le repère du cône et le repère de la caméra, et $\mathbf{K}$ la matrice de projection (associée aux paramètres intrinsèques de la caméra).

4.3.3 Discussion et généralisation

Dans ce paragraphe, nous allons aborder dans un premier temps une généralisation des résultats précédents sur les contours occultant de surfaces développables. Puis nous discuterons de l'intérêt d'utiliser des surfaces développables pour modéliser le corps humain dans le cadre du suivi du mouvement.

Généralisation Dans les paragraphes précédents, nous avons considéré dans un premier temps les surfaces développables puis nous nous sommes rapidement restreint à l'étude du cône. Dans cette partie, nous allons revenir sur le cas un peu plus général des surfaces développables. De manière générale, une surface développable est une surface qui peut être parcourue par une droite ([61]).

Etant donné cette définition, nous pouvons dériver à l'infini la forme des surfaces développables. Nous pouvons citer quelques cas particulier :

Les hélicoïdes : Ces surfaces sont générées à partir d'une hélice *3D* dont on prend la développante.

La surface a pour forme paramétrique :

$$\boldsymbol{X}(u,v) = \begin{bmatrix} a(\cos(u) - v\sin(u)) \\ a(\sin(u) + v\cos(u)) \\ c(u+v) \end{bmatrix}, \tag{4.24}$$

avec a la largeur de l'hélice, c le pas de l'hélice, u et v les paramètres de la courbe.

Les rubans de Möbius développables : De manière générale, les rubans de Möbius ne sont pas développables. Seuls quelques cas particuliers le sont.

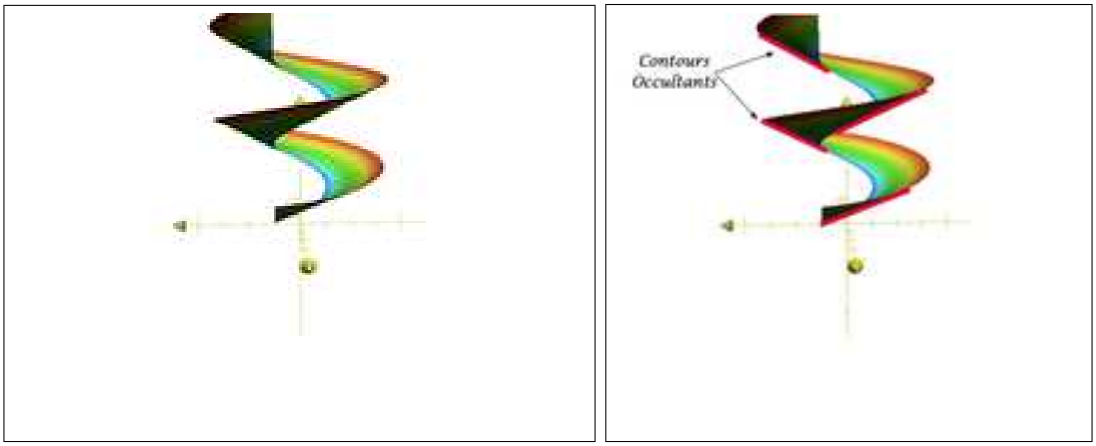


FIG. 4.11: Les contours occultant d'une hélicoïde développable se projettent dans les images sous la forme de segments de droites.

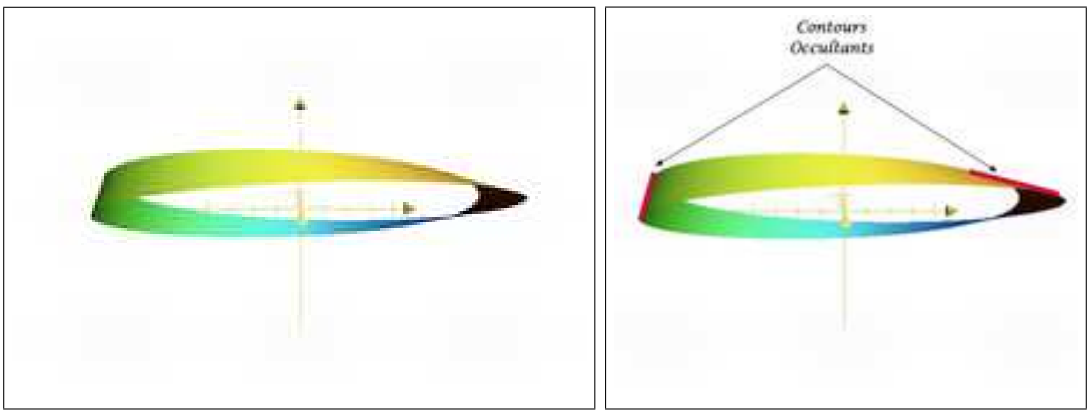


FIG. 4.12: Les contours occultant d'un ruban de Möbius développable se projettent dans les images sous la forme de segments de droites.

Ces rubans ont pour forme paramétrique :

$$\boldsymbol{X}(u,v) = \begin{bmatrix} (1+v\cos(u))\cos(2u) \\ (1+v\cos(u))\sin(2u) \\ v\sin(u) \end{bmatrix}, \tag{4.25}$$

où $v \in [0 \dots \pi]$ et $t \in [-0.2 \dots 0.2]$.

Les cônes ou cylindres généralisés : Le nombre de contours occultant peut alors être supérieur à 2. Nous donnons ici l'exemple d'un cône où la courbe directrice est une cardioïde ou encore un cône sinusoïdal.

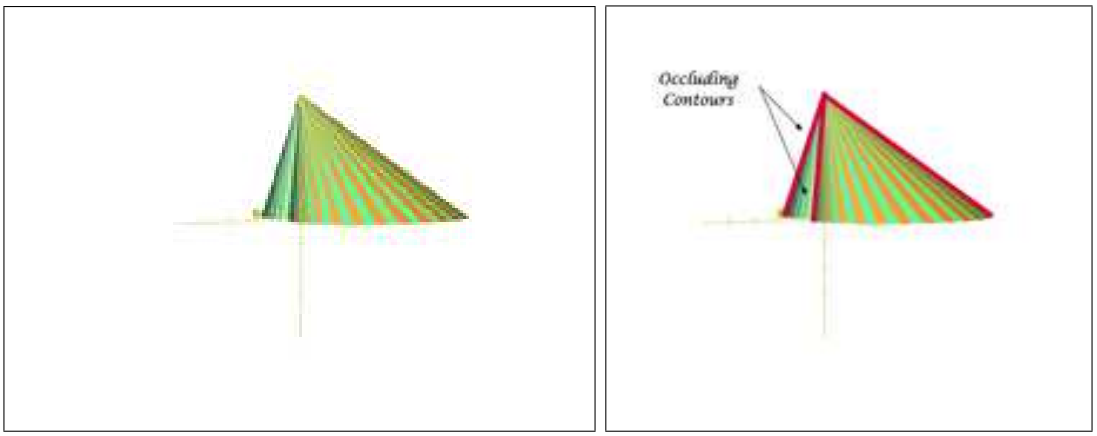


FIG. 4.13: Les contours occultant de cônes généralisés peuvent devenir complexes.

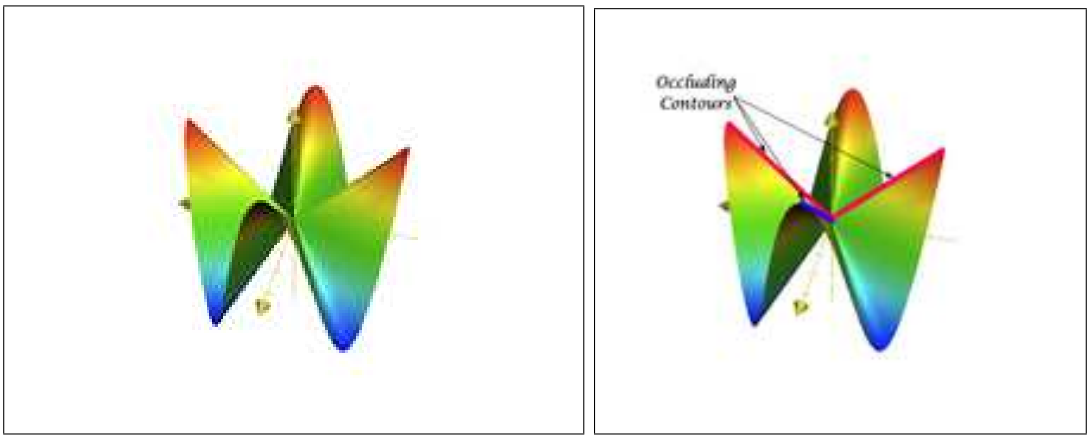


FIG. 4.14: Les contours occultant de cônes généralisés peuvent devenir complexes. Nous donnons ici l'exemple du cône sinusoïdal.

La projection de ces surfaces dans les images fait apparaître des contours extrémaux, qui sont dans tous les cas des segments de droite. Bien évidemment, ces surfaces ne rentrent pas dans le cadre de la modélisation du corps humain, cependant elles montrent

la généralisation possible du développement mathématique précédent pour le suivi d'objet à l'aide de contours occultant.

Discussion Nous pouvons nous poser la question du choix de surfaces aussi particulières que les surfaces développables pour modéliser le corps humain. En effet, les ellipsoïdes peuvent modéliser plus finement certaines des parties du corps (comme la forme des mollets ou des muscles plus généralement) et peuvent donc paraître plus adéquats pour la modélisation surfacique.

Nous avons fait un choix stratégique. En effet, les contours extrémaux engendrés par les surfaces développables sont des segments de droite tandis que ceux engendrés par d'autres surfaces peuvent être complexes. Ces segments de droite sont facilement calculables et manipulables. En effet, la connaissance de deux points du segment suffit pour complètement le déterminer. La projection des ellipsoïdes est beaucoup plus complexe (c.f. annexe B.2). Le choix du cône comme primitive représentant la plupart des parties du corps est donc d'abord un choix de simplicité.

D'autre part, nous verrons que pour estimer la pose de l'acteur, nous échantillonnons les contours occultant. Nous projetons les points dans les images et les mettons en correspondance avec les contours extraits des images. Si les contours extraits des images observées étaient des contours idéaux, c'est-à-dire deux contours extrémaux par partie du corps, alors deux points par contour du modèle seraient suffisant pour estimer la pose. Cependant, dans la réalité, les contours extraits peuvent être bruités. Donc le nombre de points utiles des contours du modèle doit être augmenté. La possibilité de choisir le nombre de points nécessaires pour effectuer le suivi va dans le sens d'une réduction des données nécessaires pour effectuer l'estimation de la pose. En effet, supposons que nous ayons six caméras, vingt et une parties du corps, et que nous prenions N points par contours occultant, il y a $2 * N * 6 * 21$ points projetés dans les images. La complexité de la minimisation dépend donc linéairement du nombre de points échantillonnés sur le contour occultant. Pour des contours plus complexes comme pour les ellipses, le nombre de points minimum est plus élevé.

Enfin, et toujours dans le sens de la réduction de données, les contours extrémaux dans les images peuvent être facilement tronqués pour prendre en compte les occultations. Plus précisément, le modèle *3D* observé par une caméra est sujet à des occultations. Les contours occultant peuvent donc être partiellement cachés par d'autres parties du corps. Pour éviter de perdre la totalité d'un contour lorsqu'une petite partie de celui-ci est cachée, nous effectuons un calcul de visibilité permettant de déterminer la partie du contour visible. Pour des segments de droites, ce calcul est simple ce qui n'est pas le cas pour des contours plus complexes.

De la visibilité des contours Nous avons exposé la méthode de projection des différentes parties du corps dans les différentes images. Cependant, nous devons faire attention aux occultations. En effet, toutes les parties du corps ne sont pas visibles dans toutes les caméras. Nous avons donc mis en place une méthode pour gérer les occultations dans les images.

La méthode utilise des techniques s'apparentant à du *ray-tracing* en graphisme. Pour faciliter le calcul de visibilité, nous utilisons les capacités de la carte graphique. Le modèle *3D* est dessiné dans un plan en utilisant les possibilités d'OpenGL [116] pour calculer les occultations. Lorsque ces parties du corps sont projetées sur le plan, il est aisé de déterminer si un point de contour d'une partie du corps est visible ou non. Pour cela, nous « dessinons » les points de contours projetés du modèle sur la projection OpenGL. Si nous définissons un attribut (une couleur) par partie du corps, alors nous vérifions que le point de contour dessiné se superpose bien avec la partie du corps ayant le même attribut (c.f. figure 4.15). Dans le cas contraire, le point de contour est invisible. Cette méthode nous permet de déterminer les points du contour occultant qui sont visibles dans la caméra. Ainsi, nous pouvons savoir si un contour est partiellement visible dans une image. Le cas échéant, la partie visible du contour est utilisée lors de la minimisation (c.f. figure 4.16).

Cette opération est très rapide, mais nécessite cependant la précaution suivante : la projection OpenGL sur le plan image utilise une rastérisation différente de celle utilisée pour le dessin des points du contour projeté. Nous devons donc faire attention à vérifier la condition de visibilité non pas sur un seul pixel mais sur un voisinage.

4.4 Le mouvement des contours apparents

L'estimation des dimensions des primitives du modèle *3D* ainsi que l'estimation du mouvement de l'acteur nécessite de calculer une erreur entre la projection du modèle *3D* et les observations dans les images. Pour pouvoir minimiser cette erreur, nous déterminons la variation de celle-ci en fonction de la variation de la projection du modèle *3D* dans les images. Le mouvement apparent dans les images des contours projetés dépend directement de la variation des paramètres du modèle *3D*. Pour le dimensionnement, les paramètres seront ceux des dimensions de chacun des cônes. Pour le suivi de l'acteur, les paramètres seront ceux de la chaîne articulaire.

Dans le paragraphe précédent, nous avons donné le paramétrage explicite des contours extrémaux dans les images en fonction des paramètres de dimension des cônes ainsi que des paramètres de la chaîne articulaire. Dans cette partie, nous allons dériver les expressions précédentes pour pouvoir exprimer le mouvement des contours extrémaux en fonction de la variation des paramètres du modèle *3D*.

Dans un premier temps, nous montrerons que le mouvement des contours extrémaux est en fait la composition de deux mouvements : le premier est lié au mouvement rigide du cône, le second est lié au mouvement relatif de la caméra par rapport au cône. Nous expliciterons alors chacun de ces mouvements en fonction des paramètres articulaires puis en fonction de la variation des dimensions du cône.

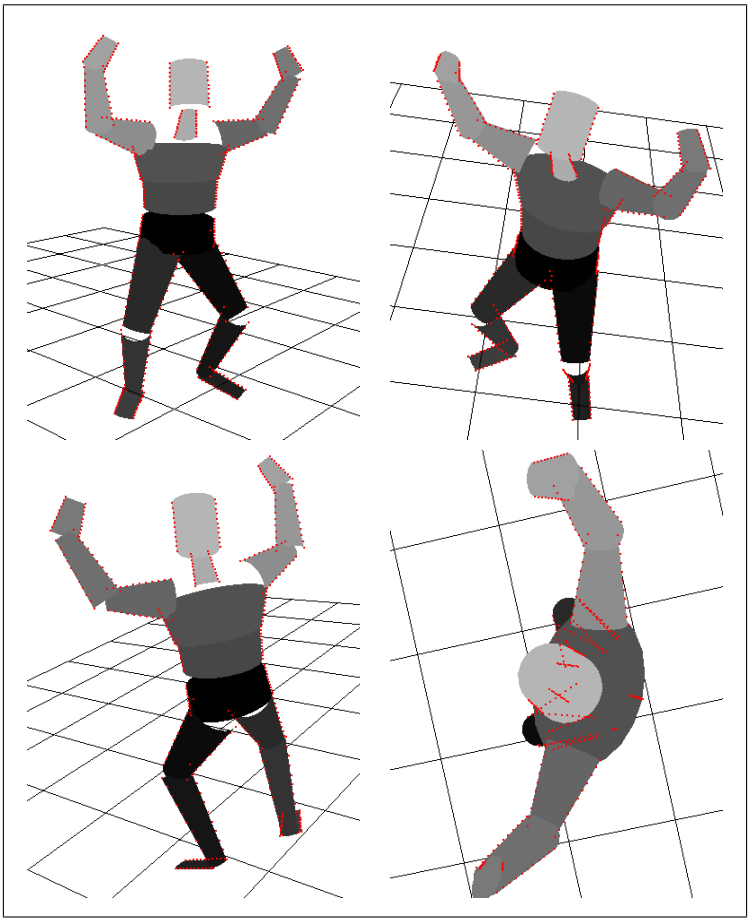


FIG. 4.15: La visibilité des contours est déterminée en utilisant les capacités de la carte graphique. Chacune des parties du modèle à une couleur. Le modèle est projeté sur la caméra en respectant les occultations (possibilité de la carte graphique). Les contours sont projetés sur cette image. Pour chaque point du contour, nous vérifions qu'il est dessiné sur la partie du corps qui lui est attitrée. Nous en déduisons alors la visibilité des contours.

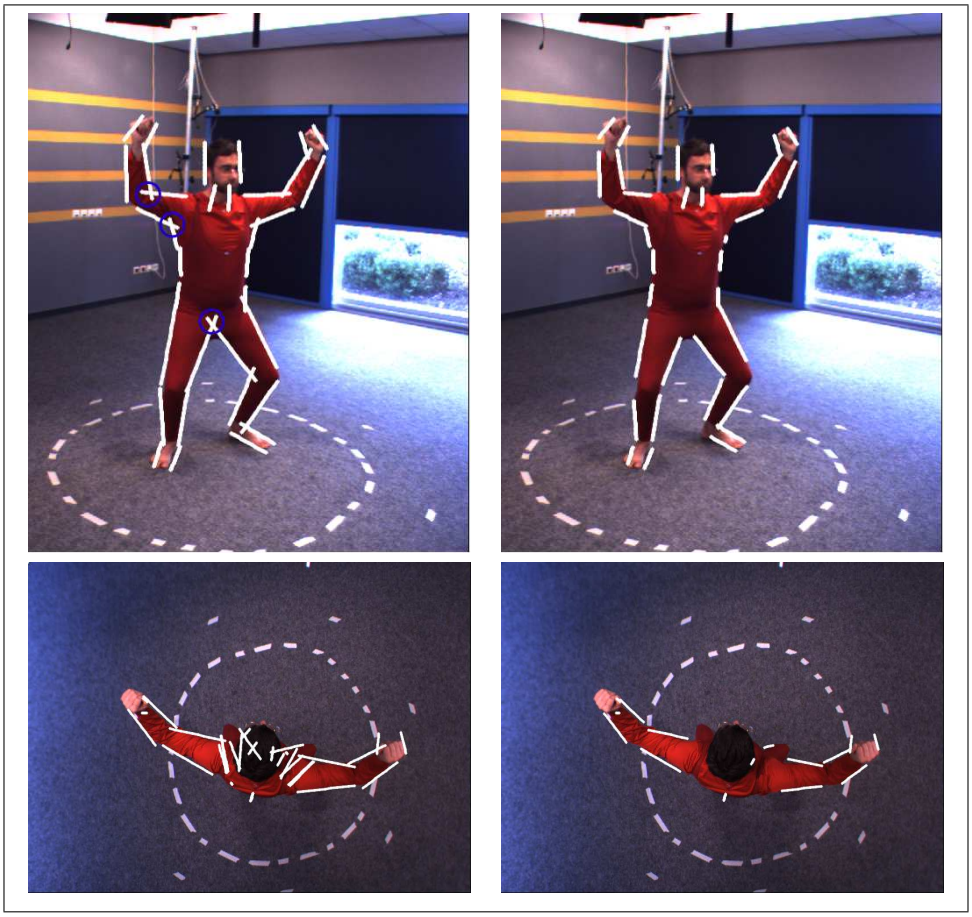


FIG. 4.16: La gestion des occultations permet, lors du suivi du mouvement, de ne pas affecter un contour modèle à une partie du corps invisible en réalité.

4.4.1 Décomposition du mouvement apparent

Pour simplifier le propos, nous considérons un cône, doté de 6 d.d.l. (3 rotations et 3 translations). Notons $\mathbf{R}$ la matrice de rotation, $\boldsymbol{t}$ le vecteur de translation, $\mathbf{\Gamma} = (\alpha_i, \alpha_j, \alpha_k, t_i, t_j, t_k)$ le vecteur des paramètres de pose de l'objet. Ces paramètres dépendent de $\mathbf{\Phi}$. De plus, nous considérons que la caméra a son centre optique confondu avec le centre du repère du monde. Pour simplifier la suite du développement, nous omettons dans un premier temps l'étude de la variation en fonction des paramètres dimensionnels. Nous y reviendrons au paragraphe 4.4.3. La matrice des paramètres intrinsèques est omise tout au long des calculs. Enfin, nous allons à nouveau restreindre notre propos au cas des surfaces développables.

Soit un point X de coordonnées $\boldsymbol{X}_{occ}(\theta, z)$ appartenant au contour occultant d'une surface développable. Nous avons vu que ses coordonnées sont déterminées par la pose de la surface dans l'espace ainsi que par la position de la caméra par rapport à la surface. Avec le paramétrage introduit avec l'équation (4.2), seul θ dépend des paramètres de pose de l'objet. Nous pouvons donc fixer $z = z_0$. Soit $\boldsymbol{x}_{occ}$ le projeté de $\boldsymbol{X}_{occ}$ sur le plan image. Nous allons expliciter la matrice Jacobienne $\mathbf{J}_{\boldsymbol{x}_{occ}}$ qui fait le lien entre la variation des paramètres articulaires du modèle *3D* et le déplacement d'un point du contour extrémal dans l'image :

$$\frac{d\boldsymbol{x}_{occ}}{dt} = \mathbf{J}_{\boldsymbol{x}_{occ}} \dot{\mathbf{\Phi}}. \tag{4.26}$$

Nous pouvons décomposer le calcul de la Jacobienne de la manière suivante :

$$\frac{d\boldsymbol{x}_{occ}}{dt} = \frac{d\boldsymbol{x}}{d\boldsymbol{X}^{\mathcal{W}}_{occ}} \frac{d\boldsymbol{X}^{\mathcal{W}}_{occ}}{d\mathbf{\Gamma}} \dot{\mathbf{\Gamma}}, \tag{4.27}$$

où $\boldsymbol{X}^{\mathcal{W}}_{occ}$ dénote les coordonnées d'un point appartenant aux contours occultant exprimées dans le repère de la caméra (confondu avec le repère du monde, d'où l'exposant $\mathcal{W}$) et nous avons vu dans le chapitre précédent que $\dot{\mathbf{\Gamma}} = \mathbf{J}_H \dot{\mathbf{\Phi}}$, où $\mathbf{\Phi}$ est le vecteur des paramètres de pose du modèle.

Nous pouvons voir qu'il s'agit en fait du produit de plusieurs matrices Jacobiennes, que nous allons expliciter dans la suite de ce chapitre.

La matrice Jacobienne image La projection du point X_{occ} (dont les coordonnées sont : $\boldsymbol{X}^{\mathcal{W}}_{occ} = (X^{\mathcal{W}}_1, X^{\mathcal{W}}_2, X^{\mathcal{W}}_3)$ dans le plan image est de la forme :

$$\boldsymbol{x}_{occ} = (x_1, x_2) = (\frac{X^{\mathcal{W}}_1}{X^{\mathcal{W}}_3}, \frac{X^{\mathcal{W}}_2}{X^{\mathcal{W}}_3}). \tag{4.28}$$

Nous pouvons alors dériver pour calculer la Jacobienne image $\mathbf{J}_I$. Elle a pour expression analytique :

$$\mathbf{J}_I = \begin{bmatrix} \frac{1}{X^{\mathcal{W}}_3} & 0 & -\frac{X^{\mathcal{W}}_1}{(X^{\mathcal{W}}_3)^2} \\ 0 & \frac{1}{X^{\mathcal{W}}_3} & -\frac{X^{\mathcal{W}}_2}{(X^{\mathcal{W}}_3)^2} \end{bmatrix}. \tag{4.29}$$

La Jacobienne image fait le lien entre le mouvement d'un point dont les coordonnées sont exprimées dans le repère caméra et le mouvement de ce point projeté sur le plan image.

Le mouvement du contour occultant Nous allons maintenant nous attarder sur le terme $\frac{d\boldsymbol{X}_{occ}^{\mathcal{W}}}{d\boldsymbol{\Gamma}}$. Il s'agit d'établir le déplacement d'un point situé sur le contour occultant, dont les coordonnées sont exprimées dans le repère du monde, en fonction des paramètres de pose de la surface ($\boldsymbol{\Gamma}$).

Dans la section précédente, nous avons établi les coordonnées d'un point du contour occultant dans le repère associé au cône. Les coordonnées de ce point dans le repère du monde sont données par l'équation (3.22) et que nous redonnons ici :

$$\boldsymbol{X}_{occ}^{\mathcal{W}} = \mathbf{R}\boldsymbol{X}_{occ} + \boldsymbol{t}. \tag{4.30}$$

L'expression de la vitesse du point X du contour occultant soumis au déplacement rigide de la surface s'obtient en dérivant l'équation précédente :

$$\dot{\boldsymbol{X}}_{occ}^{\mathcal{W}} = \dot{\mathbf{R}}\boldsymbol{X}_{occ} + \mathbf{R}\dot{\boldsymbol{X}}_{occ} + \dot{\boldsymbol{t}}. \tag{4.31}$$

Par abus de notation nous notons $(\dot{-})$ la dérivée par rapport à n'importe quel paramètre de pose du cône.

Nous voyons dans cette équation que le mouvement du point de contour est la somme de deux mouvements :

- **Le mouvement rigide** ($\dot{\mathbf{R}}\boldsymbol{X}_{occ} + \dot{\boldsymbol{t}}$) de la surface au point $\boldsymbol{X}_{occ}$.
- **Le mouvement de glissement** du point sur la surface ($\mathbf{R}\dot{\boldsymbol{X}}_{occ}$).

Nous allons rappeler brièvement les formules introduites dans le chapitre précédent pour le mouvement rigide. Puis nous rentrerons plus en détails dans la description du mouvement de glissement du point sur la surface du cône.

4.4.2 Variation des paramètres de pose

Nous allons expliciter de manière analytique chacun des termes du mouvement lié à la variation des paramètres articulaires. Pour simplifier les écritures, nous omettrons l'indice $_{occ}$ lorsque les notations ne seront pas ambiguës.

Le mouvement rigide Nous l'avons longuement décrit dans le chapitre précédent au paragraphe 3.3. La Jacobienne est de la forme :

$$\mathbf{J}_{rigid} = \begin{bmatrix} [\boldsymbol{t} - \boldsymbol{X}^{\mathcal{W}}]_{\times} & \mathbf{I}_{3\times 3} \end{bmatrix}. \tag{4.32}$$

Le mouvement de glissement Beaucoup de travaux ont été menés sur l'établissement du lien entre les surfaces et leurs observations dans les images (thématique du *shape from silhouette*). Cependant peu de travaux ont abordé l'étude du mouvement des contours dans les images en fonction du mouvement de la surface. Ce problème est lié à celui de l'étude différentielle des courbes et surfaces ([87] et [50] aux chapitres 19 et 20). Ces deux livres ne traitent que du cas où les contours sont observés sous une projection orthographique. Nous nous intéressons au cas de la projection perspective. [88] traite des contours occultant. [126] propose d'utiliser les contours occultant pour suivre des surfaces dans les images. Les auteurs affirment que le mouvement des contours apparent dans les images ne dépend pas du mouvement de glissement. Au contraire, nous pensons que prendre en compte ce mouvement permet de mieux rendre compte du mouvement réel des contours apparents (c.f. paragraphe 4.4.4).

Dans ce paragraphe, nous allons expliciter la dérivation de $\boldsymbol{X}^{\mathcal{W}}$ par rapport aux paramètres de pose. Pour pouvoir calculer le mouvement de glissement, nous allons calculer l'équation aux dérivées partielles suivante :

$$\frac{d\boldsymbol{X}^{\mathcal{W}}}{d\boldsymbol{\Gamma}} = \boldsymbol{X}_{\theta}^{\mathcal{W}} \frac{d\theta}{d\boldsymbol{\Gamma}} + \boldsymbol{X}_{z}^{\mathcal{W}} \frac{dz}{d\boldsymbol{\Gamma}}. \tag{4.33}$$

Les contours occultant des cônes sont des segments de droite. Ils sont paramétrés par θ et z. Cependant, nous avons pu voir que seul θ dépend des paramètres de pose de la chaîne articulaire. Donc $\frac{dz}{d\boldsymbol{\Gamma}} = 0$. Le terme $\boldsymbol{X}_{\theta}^{\mathcal{W}} = \frac{d\boldsymbol{X}^{\mathcal{W}}}{d\theta}$ est facilement calculable. Nous allons donc établir l'expression analytique de $\frac{d\theta}{d\boldsymbol{\Gamma}}$. Pour cela, reconsidérons la contrainte d'appartenance d'un point de la surface à un contour occultant (4.12) :

$$(\boldsymbol{X} + \mathbf{R}^{\top}(\boldsymbol{t} - \boldsymbol{C}))^{\top}\boldsymbol{n} = 0. \tag{4.34}$$

La dérivation de cette équation par rapport à un paramètre du mouvement donne :

$$(\boldsymbol{X} + \mathbf{R}^{\top}(\boldsymbol{t} - \boldsymbol{C}))^{\top}\dot{\boldsymbol{n}} = -(\dot{\boldsymbol{X}} + \dot{\mathbf{R}}^{\top}(\boldsymbol{t} - \boldsymbol{C}) + \mathbf{R}^{\top}\dot{\boldsymbol{t}})^{\top}\boldsymbol{n} \tag{4.35}$$

Nous pouvons modifier et simplifier cette équation en considérant les points suivants :
- La vitesse d'un point sur la surface est tangente à cette surface, nous avons donc : $\dot{\boldsymbol{X}}^{\top}\boldsymbol{n} = 0$
- Avec l'équation (3.39) introduite au paragraphe 3.4.1, nous pouvons écrire : $\dot{\mathbf{R}}^{\top} = -\mathbf{R}^{\top}[\boldsymbol{\omega}]_{\times}$.
- Nous avons $\dot{\boldsymbol{n}} = \boldsymbol{n}_{\theta}\dot{\theta}$

Nous avons donc :

$$\begin{aligned}(\boldsymbol{X} + \mathbf{R}^{\top}(\boldsymbol{t} - \boldsymbol{C}))^{\top}\boldsymbol{n}_{\theta}\dot{\theta} &= -(-\mathbf{R}^{\top}[\boldsymbol{\omega}]_{\times}(\boldsymbol{t} - \boldsymbol{C}) + \mathbf{R}^{\top}\dot{\boldsymbol{t}})^{\top}\boldsymbol{n} \\ &= ([\boldsymbol{\omega}]_{\times}(\boldsymbol{t} - \boldsymbol{C}) - \dot{\boldsymbol{t}})^{\top}\mathbf{R}\boldsymbol{n}\end{aligned} \tag{4.36}$$

En utilisant la réécriture introduite dans (3.47) :

$$[\boldsymbol{\omega}]_{\times}(\boldsymbol{t} - \boldsymbol{C}) - \dot{\boldsymbol{t}} = \begin{bmatrix} [\boldsymbol{C} - \boldsymbol{t}]_{\times} & -\mathbf{I}_{3\times 3} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}, \tag{4.37}$$

et le fait que $([\boldsymbol{\omega}]_\times(\boldsymbol{t}-\boldsymbol{C})-\dot{\boldsymbol{t}})^\top\mathbf{R}\boldsymbol{n} = (\mathbf{R}\boldsymbol{n})^\top([\boldsymbol{\omega}]_\times(\boldsymbol{t}-\boldsymbol{C})-\dot{\boldsymbol{t}})$ (puisqu'il s'agit d'un scalaire), nous avons :

$$\boxed{\dot{\theta} = \frac{(\mathbf{R}\boldsymbol{n})^\top \begin{bmatrix} [\boldsymbol{C}-\boldsymbol{t}]_\times & -\mathbf{I}_{3\times 3} \end{bmatrix}}{(\boldsymbol{X}+\mathbf{R}^\top(\boldsymbol{t}-\boldsymbol{C}))^\top\boldsymbol{n}_\theta} \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}}, \tag{4.38}$$

tant que $(\boldsymbol{X}+\mathbf{R}^\top(\boldsymbol{t}-\boldsymbol{C}))^\top\boldsymbol{n}_\theta$ est non nul. $(\boldsymbol{X}+\mathbf{R}^\top(\boldsymbol{t}-\boldsymbol{C}))^\top\boldsymbol{n}_\theta$ tend vers 0 si la courbure de l'objet tend vers 0 ($\boldsymbol{n}_\theta \to 0$).

L'équation (4.38) établit le lien entre la vitesse de déplacement rigide de la surface et la vitesse de déplacement d'un point du contour occultant. Nous pouvons la reformuler de la manière suivante :

$$\dot{\theta} = \mathbf{J}_{sliding}(\theta, \mathbf{R}, \boldsymbol{t}, \boldsymbol{C}) \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix} \tag{4.39}$$

La vitesse de glissement d'un point du contour sur l'objet est donc fonction :
- du point en lequel la vitesse est calculée,
- de la position relative de la caméra par rapport à la surface,
- du mouvement rigide de la surface.

Synthèse : Nous pouvons donc écrire la vitesse de déplacement d'un point du contour occultant de la manière suivante :

$$\dot{\boldsymbol{X}}^{\mathcal{W}}_{occ} = (\mathbf{A}+\mathbf{B}) \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}, \tag{4.40}$$

avec $\mathbf{A} = \mathbf{J}_{rigid}$ et $\mathbf{B} = \mathbf{R}\boldsymbol{X}_\theta\mathbf{J}_{sliding}$.

Le mouvement des contours extrémaux Nous avons étudié le mouvement du contour occultant, c'est-à-dire le mouvement du contour glissant sur la surface *3D*. Nous pouvons maintenant évaluer la vitesse de déplacement du contour observé dans l'image.

Nous avons :

$$\boxed{\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A}+\mathbf{B}) \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}}. \tag{4.41}$$

La cas de la chaîne cinématique Nous avons étudié le cas d'une surface développable observée par une caméra et soumise à un déplacement rigide. L'extension au cas de la chaîne cinématique se fait en utilisant les résultats du premier chapitre et plus particulièrement l'équation (3.86). Ainsi nous avons, pour une surface dont le mouvement est paramétré par celui d'une chaîne articulaire :

$$\boxed{\dot{\boldsymbol{x}} = \mathbf{J}_I(\mathbf{A}+\mathbf{B})\mathbf{J}_H\dot{\boldsymbol{\Phi}}}, \tag{4.42}$$

où $\mathbf{J}_H$ est définie par l'équation (3.86).

De la même manière que pour le calcul de la projection des points dans l'image, il n'est pas nécessaire de calculer la Jacobienne des contours occultant pour tous les points du contour. En effet, la variation de la vitesse le long du contour dépend linéairement de z. Il suffit donc de calculer la Jacobienne pour deux points du contour occultant, les autres se déduisant par combinaison linéaire. Cependant, cette simplification n'est pas vraie pour la Jacobienne des points dans l'image car la projection n'est pas une opération linéaire. Nous calculons donc la Jacobienne de tous les points *3D*, et faisons la projection ensuite.

4.4.3 Variation des paramètres de dimension

Une des premières phases de la capture du mouvement consiste à dimensionner le modèle *3D* (nous le verrons au chapitre suivant). Pour dimensionner le modèle, nous projetons les primitives du modèle dans les images sous la forme de contours extrémaux et modifions les paramètres dimensionnels (de chacun des cônes) pour que les contours projetés correspondent au mieux avec les contours observés dans les images. Nous allons maintenant étudier le mouvement apparent des contours en fonction de la variations des paramètres dimensionnels du cône.

Pour simplifier le propos, nous supposons que la hauteur du cône est fixée par la longueur des éléments du squelette. L'évaluation de la hauteur du cône est effectuée lors de la phase d'adaptation de la chaîne articulaire à l'acteur, ce dont nous discuterons au chapitre 5. Les seuls paramètres à optimiser sont alors les demis grand et petit axes de la base des cônes (a_b et b_b) ainsi que le coefficient $k = -\frac{1}{l}$ caractérisant l'ouverture du cône. Nous allons montrer que nous pouvons écrire la relation sous la forme :

$$\frac{d\boldsymbol{X}_{occ}}{dt} = \mathbf{J}_{dim}\dot{\boldsymbol{\Sigma}}, \tag{4.43}$$

où $\boldsymbol{\Sigma} = (a_b, b_b, k)$.

Nous avons vu que le contour occultant peut être paramétré de la manière suivante :

$$\boldsymbol{X}(\theta, z) = \begin{bmatrix} a(1+kz)\cos(\theta) \\ b(1+kz)\sin(\theta) \\ z \end{bmatrix} \tag{4.44}$$

où θ dépend des paramètres de pose du cône. Nous avons pu voir que θ dépendait aussi des dimensions du cône. En effet, dans l'équation (4.22) (les solutions du polynôme du second degré permettant de calculer θ), chacun des coefficients dépend des paramètres du cône. D'autre part, nous avons vu que les contours occultant sont des segments de droite et donc seul θ dépend des paramètres dimensionnels du cône. Pour pouvoir établir la variation d'un point du contour dans l'image en fonction de la variation de ces paramètres, la difficulté principale est de calculer la variation de θ en fonction de ces paramètres. Nous allons maintenant déterminer cette relation. Soit μ l'un des paramètres, nous allons expliciter $\frac{d\theta}{d\mu}$.

Comme pour l'estimation de la variation en fonction des paramètres articulaires, nous utilisons la contrainte d'appartenance d'un point au contour occultant :

$$(\boldsymbol{X} + \mathbf{R}^\top(\boldsymbol{t} - \boldsymbol{C}))^\top \boldsymbol{n} = 0. \tag{4.45}$$

$\mathbf{R}$, $\mathbf{t}$ et $\boldsymbol{C}$ ne dépendent pas des dimensions des cônes. On peut donc simplifier l'expression en posant $\mathbf{R} = \mathbf{I}$ et $\boldsymbol{t} = \mathbf{0}$. Cependant, les coordonnées de la caméra seront exprimées dans le repère du cône. Nous obtenons donc comme nouvelle expression de la contrainte :

$$(\boldsymbol{X} - \boldsymbol{C}_c)^\top \boldsymbol{n} = 0, \tag{4.46}$$

où $\boldsymbol{C}_c$ est le vecteur des coordonnées de la caméra exprimées dans le repère du cône. En écrivant l'équation aux dérivées partielles, nous obtenons :

$$\frac{\partial}{\partial \mu}(\boldsymbol{X} - \boldsymbol{C}_c)^\top \boldsymbol{n} + \frac{\partial}{\partial \theta}\left((\boldsymbol{X} - \boldsymbol{C}_c)^\top \boldsymbol{n}\right)\frac{\partial \theta}{\partial \mu} = 0. \tag{4.47}$$

Or la vitesse de déplacement d'un point sur la surface est tangentielle à celle-ci. Nous pouvons alors simplifier l'équation (4.47) avec $\frac{d\boldsymbol{X}_{occ}}{d\theta}^\top \boldsymbol{n} = 0$. D'autre part, les coordonnées de la caméra ne dépendant pas de μ, nous avons :

$$\theta_\mu = -\frac{\frac{\partial}{\partial \mu}((\boldsymbol{X} - \boldsymbol{C}_c)^\top \boldsymbol{n})}{(\boldsymbol{X} - \boldsymbol{C}_c)^\top \boldsymbol{n}_\theta} \tag{4.48}$$

Nous pouvons, maintenant, donner explicitement la variation de θ en fonction des paramètres. Nous avons un dénominateur dont la valeur ne dépend pas du paramètre de dérivation et a donc pour expression :

$$s = (\boldsymbol{X} - \boldsymbol{C})^\top \boldsymbol{n}_\theta = \frac{1}{2}ab(1 + kz)\sin(2\theta) + c_1 b sin(\theta) - c_2 a\cos(\theta), \tag{4.49}$$

où c_1 et c_2 sont 2 des coordonnées de $\boldsymbol{C}_c = (c_1, c_2, c_3)$.

Le numérateur dépend du paramètre de dérivation. Nous avons :

$$\frac{\partial}{\partial a}((\boldsymbol{X} - \boldsymbol{C})^\top \boldsymbol{n}) = b - (c_2\sin(\theta) - c_3 bk) \tag{4.50}$$

$$\frac{\partial}{\partial b}((\boldsymbol{X} - \boldsymbol{C})^\top \boldsymbol{n}) = a - (c_1\cos(\theta) - c_3 ak) \tag{4.51}$$

$$\frac{\partial}{\partial k}((\boldsymbol{X} - \boldsymbol{C})^\top \boldsymbol{n}) = c_3 ab \tag{4.52}$$

Nous pouvons remarquer que la variation de l'angle n'est pas nulle si nous faisons varier k. Ceci est dû au fait que l'ouverture du cône est lié à k. Et si l'ouverture du cône varie, les contours glissent à sa surface.

4.4.4 Analyse géométrique du mouvement et discussions

Nous avons vu que le mouvement d'un point contour est la somme de deux termes : le premier lié au mouvement rigide de la surface et le second lié au mouvement relatif de la caméra par rapport au cône. Nous pouvons maintenant nous poser la question de l'importance de ce second terme par rapport au premier. [126] propose une méthode de suivi d'objets à l'aide des contours occultant. Les auteurs affirment que le mouvement de glissement du contour sur la surface n'est pas visible dans les images. En effet, le vecteur vitesse de glissement dans l'image est tangent au contour apparent (dans l'image). Les auteurs ne prennent donc pas en compte ce mouvement, puisqu'il n'est pas visible. Contrairement à ces auteurs, nous pensons que le mouvement lié au glissement du contour sur la surface, bien que non prédominant, joue un rôle important dans la convergence de l'algorithme de suivi du mouvement.

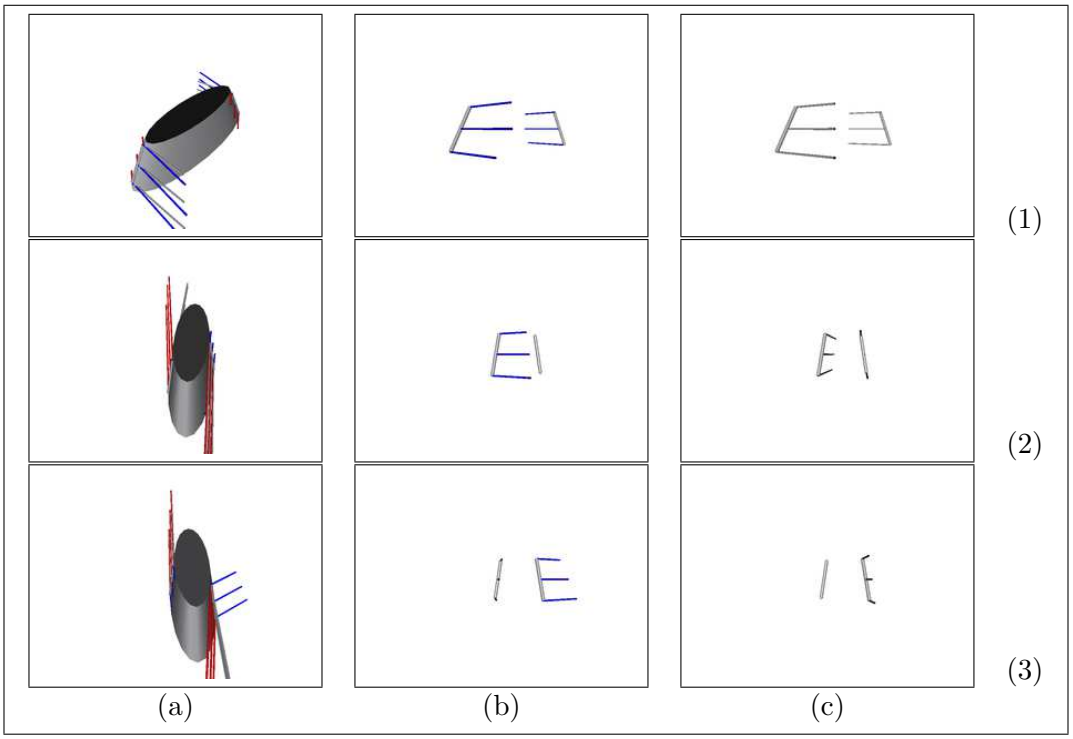


FIG. 4.17: Le mouvement apparent du contour et le mouvement rigide dans l'image sont différents. (a) est une vue des vecteurs du mouvement avec en rouge le mouvement de glissement, en bleu le mouvement rigide et en gris la somme des deux mouvements à différents instants. (b) est la projection du mouvement rigide. Enfin (c) est la projection du mouvement réel du contour dans les images.

Afin de représenter les deux mouvements sur un exemple simple, nous avons simulé le mouvement apparent d'un cône elliptique dans la figure 4.17. Dans la position (1), le mouvement rigide prédit correctement le mouvement observé mais dans les positions (2) et (3), les différences sont très importantes, à la fois en norme et en direction. En

effet, le seul mouvement rigide ne rend pas compte du mouvement réel observé (c.f. figure 4.17). Le mouvement de glissement peut devenir prédominant sur la surface *3D*. Cette prédominance s'observe notamment lorsque le contour est situé sur la portion de surface où le rayon de courbure est grand. Le mouvement apparent réel dans l'image est alors faible comparé au mouvement rigide (c.f. figure 4.17-(2) et (3)). D'un autre coté, le mouvement de glissement devient négligeable sur les portions de surface où le rayon de courbure est faible. Le mouvement apparent et le mouvement rigide sont alors quasiment confondus (c.f. figure 4.17-(1)).

La différence que nous observons entre le mouvement rigide et le mouvement total permet d'accélérer l'estimation des paramètres de pose lors du suivi du mouvement.

Chapitre 5

Le suivi du mouvement

Sommaire

Résumé

Nous abordons le problème de l'extraction et de la mise en correspondance des données. Nous établissons les fonctions permettant de mesurer l'écart entre les observation et les prédictions données par le modèle, que ce soit pour l'utilisation de la couleur ou dans le cadre de l'utilisation des contours. Pour le cas de la couleur, nous donnons l'expression analytique de la fonction d'erreur. En ce qui concerne l'utilisation des contours, nous utilisons la distance de Hausdorff, avec une transformée en distance comme métrique associée. **La description explicite du calcul de la distance de Hausdorff constitue la seconde contribution de cette thèse.** A notre connaissance, une telle description n'existe pas.

Ensuite, nous abordons trois des points essentiels du processus de capture du mouvement : la capture de l'acteur, l'initialisation du mouvement et la capture du mouvement. Nous montrons que si chacun de ces points peut utiliser des données différentes, nous pouvons utiliser le même algorithme de minimisation non linéaire (algorithme de Levenberg-Marquardt) pour effectuer chacune de ces étapes.

La capture de l'acteur constitue la première étape. L'objectif est de dimensionner le modèle *3D* pour que celui-ci corresponde au mieux à l'acteur filmé. Pour effectuer le dimensionnement, nous utilisons la couleur. Pour un cône donné, nous modifions ses dimensions de sorte à ce que les contours de celui-ci projetés dans l'image approchent au mieux la ligne de rupture de couleur entre l'acteur et le fond de l'image. Il s'agit d'une méthode locale qui nécessite que la pose du modèle *3D* soit correcte. Pour satisfaire cette contrainte, nous procédons auparavant à l'initialisation semi-automatique du squelette articulé.

La seconde étape est l'initialisation de la séquence de suivi du mouvement. Pour chacune des séquences de capture du mouvement, nous devons initialiser la pose du modèle. Le problème posé par cette initialisation est que la pose du modèle peut être assez éloignée de la pose initiale de l'acteur. Pour effectuer l'initialisation, nous procédons donc avec une approche hiérarchique. Nous commençons par estimer la pose globale de l'acteur en ne laissant libre que le mouvement rigide du pelvis. Puis, au fur et à mesure de la minimisation, nous libérons les différents degrés de liberté (d.d.l.) du squelette.

Enfin, la dernière étape est la capture du mouvement. Pour effectuer le suivi du mouvement, nous pouvons utiliser la couleur, les silhouettes et/ou les contours. Alors que les détecteurs de contours standards utilisent des images en niveaux de gris, nous montrons que l'utilisation des images couleur améliore la détection des contours. Cependant, beaucoup de contours parasites et donc gênants pour la minimisation apparaissent lors de cette détection. Pour palier à cette difficulté, nous proposons une méthode de détection de contours permettant de s'affranchir des contours parasites pour ne retenir que les contours utiles à la minimisation. Nous utilisons une détection de contours anisotropique dont l'orientation est donnée par le modèle. **Ce point constitue la troisième contribution majeure.**

Introduction au chapitre

Dans les chapitres précédents, nous avons présenté le modèle *3D* que nous utilisons pour effectuer le suivi. D'une part , nous avons modélisé la chaîne cinématique du corps humain et d'autre part nous avons présenté le modèle volumique que nous utilisons. De plus, nous nous sommes attardé sur la modélisation analytique de la projection du modèle dans les images. Nous avons explicité le mouvement du contour apparent du modèle en fonction de la variation des paramètres articulaires et dimensionnels du modèle *3D*. Nous abordons le sujet principal de cette thèse qui est le problème du suivi du mouvement.

La capture du mouvement d'un acteur s'effectue en trois étapes :
- Dimensionnement du modèle *3D*,
- Initialisation du suivi,
- Le suivi

Pour réaliser ces trois étapes nous disposons des images acquises à l'aide de plusieurs caméras calibrées ainsi que d'un modèle *3D* représentant la morphologie humaine. Pour chacune des étapes, l'objectif est de faire en sorte que le modèle *3D* corresponde aux observations. La première étape permet d'ajuster les dimensions du modèle *3D* pour qu'il corresponde à la morphologie de l'acteur dans une pose de référence. La seconde étape permet de positionner le modèle sur la première image de la séquence vidéo pour que la pose du modèle corresponde à celle observée. Enfin, la dernière étape permet de mettre à jour la pose au cours du temps à partir de cette initialisation.

Pour chacune de ces trois étapes, il y a deux problèmes à résoudre. D'une part, nous devons mettre en correspondance les données observées dans les images avec les données du modèle *3D*. D'autre part, nous devons estimer les paramètres du modèle pour que l'écart entre l'observation et le modèle soit minimal. Alors que la phase de mise en correspondance diffère pour chacune des étapes, nous allons voir que la phase d'estimation des paramètres est ramenée à un problème de minimisation. Etant donné l'ensemble des observations $\mathcal{Y}$ et les prédictions $\mathcal{X}$ données par le modèle et qui dépendent des paramètres du modèle $\boldsymbol{\Upsilon}$, nous voulons résoudre le problème suivant :

$$\min_{\boldsymbol{\Upsilon}} E(\mathcal{X}, \mathcal{Y}), \tag{5.1}$$

où E est une mesure de l'erreur entre l'estimation et l'observation, $\boldsymbol{\Upsilon}$ est la concaténation des paramètres de pose de la chaîne articulaire $\boldsymbol{\Phi}$ et des paramètres dimensionnels de tous les cônes $\boldsymbol{\Sigma}$.

Nous commençons donc ce chapitre par une description des primitives des images que nous utilisons pour mettre en correspondance le modèle *3D* avec les observations. Nous développons l'utilisation des silhouettes, des contours extraits avec des méthodes standards ainsi qu'une méthode de détection de contours utilisant la connaissance a priori du modèle. Ce dernier point constitue la seconde contribution de cette thèse. Enfin, nous décrivons l'utilisation de la couleur dans les images pour effectuer la capture de la couleur.

Dans une deuxième partie, nous abordons la mise en correspondance des observations avec le modèle *3D*. Nous établissons alors l'expression analytique des erreurs entre l'observation et le modèle. Pour calculer ces erreurs, nous utilisons les développements introduits au chapitre 4 et plus précisément le paramétrage des contours apparents. Avec les silhouettes ou les contours, nous utilisons la distance de Hausdorff comme mesure de l'erreur. Nous montrons que nous pouvons rendre cette distance continue et dérivable. Nous montrons que de par le choix de notre modèle *3D*, nous pouvons garder l'aspect symétrique de la distance. Enfin, nous explicitons le calcul de l'erreur, dans le cas où nous utilisons la couleur.

Dans la partie 5.3, nous développons l'estimation des paramètres du modèle *3D* pour la capture du mouvement. Il s'agit d'un problème de moindres carrés, que nous résolvons à l'aide de l'*algorithme de Levenberg-Marquardt*.

Nous étendons l'algorithme de base pour le cas de l'initialisation dans la section 5.4. Enfin, nous traitons du dimensionnement du modèle *3D* dans la section 5.5.

5.1 Extraction de primitives

Nous avons pu voir dans l'état de l'art que plusieurs techniques sont proposées pour effectuer le suivi du mouvement humain à l'aide de plusieurs caméras.

Nous avons justifié le fait que notre méthode s'appuie non pas sur la reconstruction *3D* mais sur la projection du modèle *3D* dans les images. Pour effectuer la comparaison entre le modèle projeté et les observations, nous pouvons utiliser la couleur, des points caractéristiques, les silhouettes ou encore les contours extraits des images.

Nous allons aborder dans cette partie les données images que nous utilisons pour effectuer le suivi du mouvement.

Les données images sont par nature des données bruitées. Que ce soit lié au capteur image ou à l'environnement d'acquisition, les sources de perturbation des acquisitions sont nombreuses. L'extraction des données utiles pour l'estimation du mouvement en est d'autant plus compliquée. Nous devons faire la part entre les informations ayant traits à l'acteur de celles liées à l'environnement d'acquisition. Il existe beaucoup de méthodes pour séparer les sources d'information. Pour la capture du mouvement, la technique la plus souvent utilisée est la soustraction de fond, permettant de différencier l'acteur du reste de l'image. Une seconde méthode consiste à effectuer un apprentissage de l'acteur et donc de le localiser dans les images de la séquence. Nous avons décidé d'utiliser la soustraction de fond d'image, qui permet d'être plus robuste aux changements d'apparence de l'acteur au cours de la séquence vidéo. Cependant ces techniques de soustraction sont très dépendantes des conditions dans lesquelles la capture du mouvement est effectuée. En effet, ces méthodes sont sensibles aux conditions d'illumination et ne sont pas utilisables dans le cas d'un fond d'image non statique. De plus, les silhouettes ne renseignent pas de manière optimale sur la position de chacune des parties du corps. Donc, nous utilisons aussi les contours extraits dans les images. Nous verrons

comment nous pouvons allier l'utilisation des silhouettes et des contours pour rendre le suivi du mouvement robuste.

5.1.1 Les silhouettes

L'utilisation des silhouettes est très répandue car elles constituent une information fiable sur la localisation de l'acteur dans les images. Pour effectuer l'extraction des silhouettes nous utilisons une technique de soustraction de fond standard utilisant une mixture de gaussienne. L'algorithme de soustraction de fond permet d'obtenir une carte de probabilité de présence de nouvel élément non appris dans l'image. A partir de cette carte, nous utilisons les *Graph-Cuts* pour extraire la silhouette de l'acteur.

L'utilisation des silhouettes pour effectuer le suivi du mouvement pose deux difficultés majeures :

- Un problème inhérent à l'extraction de la silhouette vient du fait que le fond doit être statique au cours de la prise de vue. En effet, toute variation du fond perturbera l'extraction de l'acteur. Pour des variations comme les changements de luminosité, le modèle du fond peut être mis à jour au cours du temps. Par contre pour des fonds d'image dynamiques, la mise à jour ne peut être faite et donc la soustraction de fond n'est pas utilisable.
- La seconde difficulté est liée à la capture du mouvement. En effet, la silhouette ne contient pas toutes les informations nécessaires pour effectuer le suivi du mouvement. Prenons, par exemple, le cas où l'acteur pose son bras le long du torse : les silhouettes ne permettent pas de distinguer le bras du corps (c.f. illustration 5.1).

En ce qui concerne le second désavantage, nous pouvons considérer que nous utilisons plusieurs caméras pour le suivi du mouvement et qu'il existe toujours des vues dans lesquelles les parties du corps seront distinguables. Cette remarque est souvent vraie sauf dans le cas où deux parties du corps sont très proches. La simple utilisation des silhouettes ne suffit donc pas à effectuer la capture du mouvement de tout type de mouvement.

5.1.2 Les contours

Pour palier aux problèmes des silhouettes, nous avons décider d'utiliser les contours. Ces derniers permettent d'obtenir une information beaucoup plus riche sur la pose de l'acteur.

Dans le cadre de la capture du mouvement, un détecteur de contour idéale serait un détecteur de contours extrémaux de l'acteur, c'est-à-dire les contours délimitant chacune des partie du corps. Certains travaux proposent des méthodes d'extraction de contours permettant d'extraire des contours « naturels » dans les images. Martin *et al.* dans [98] proposent d'extraire les contours d'une image en utilisant un apprentissage préalable de contours extraits par des utilisateurs. Un résultat est présenté sur la figure 5.2. Les résultats obtenus sont très bons mais nécessitent des temps de calcul très longs. Pour l'image 5.2, avec un processeur Intel Core Duo® 2Ghz et 1Go de Ram, il nous a fallu

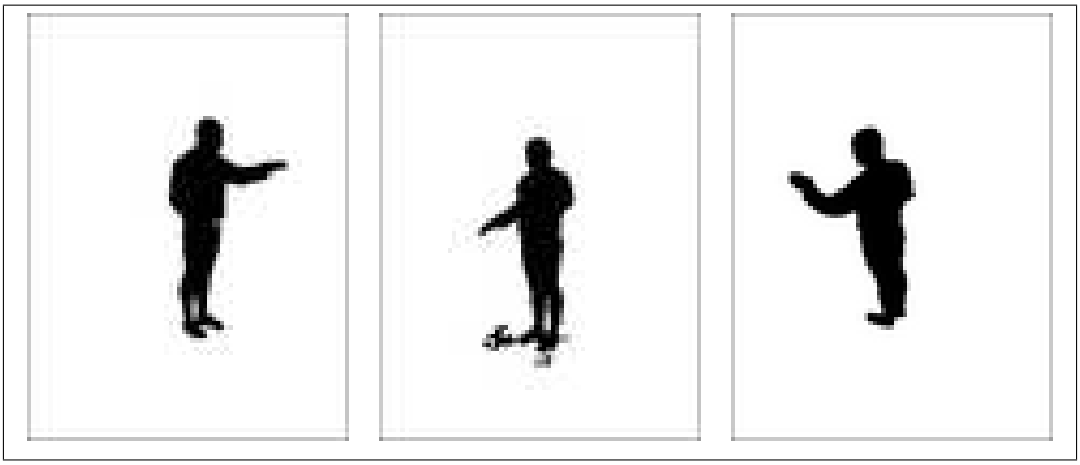


FIG. 5.1: Les silhouettes extraites de plusieurs points de vues ne permettent pas de distinguer l'ensemble des parties du corps. Dans l'exemple donné ici, un bras est situé le long du corps.

3 min 12s, ce qui est trop long pour les applications que nous visons. Plus récemment, Dollar *et al.* dans [43], proposent une méthode de segmentation plus rapide avec de meilleurs résultats. Ces méthodes d'extraction de contours ne s'affranchissent pas de la soustraction de fond puisque tous les contours dans l'image sont détectés. Ce ne sont pas des détecteurs de contours extrémaux. Dans la pratique, ces derniers n'existent pas.

Dans la suite de ce paragraphe, nous abordons différentes méthodes de détection de contours que nous avons mises en place.

5.1.2.1 Méthode Standard

Les détecteurs de contours les plus couramment utilisés prennent en entrée des images en niveaux de gris. Ils permettent de détecter des changements d'intensité. Parmi eux, nous pouvons citer les détecteurs de Prewitt, Sobel, Laplace [57] ou encore le filtre de Canny [22]. Ces détecteurs étudient la structure différentielle locale d'images. Ces détecteurs souffrent d'un problème majeur : ils ne détectent pas les contours isoluminants. C'est-à-dire qu'un contour séparant deux objets ayant la même luminosité ne sera pas détecté. Dans le cas de la capture du mouvement, si les éclairages sont correctement répartis, ces contours apparaissent souvent (des exemples de contours isoluminants sont donnés sur la première et seconde lignes de la figure 5.3). Nous devons donc modifier le détecteur pour pouvoir extraire tous les contours de l'image.

5.1.2.2 Filtrage de Canny adapté à la couleur

De la même manière que pour les filtres standards, il s'agit d'étudier la structure différentielle des images. Cependant, les filtres standards utilisent la luminance

FIG. 5.2: Résultats obtenus à l'aide du détecteur de contours de Berkley.

et perdent donc l'information sur la chromaticité, c'est-à-dire sur le changement de couleur dans les images. L'étude de la structure différentielle de chacun des canaux de couleur permet de rendre compte de toute l'information.

Les méthodes standard de détection de contours utilisent généralement les directions et les normes des gradients pour effectuer le filtrage, ce qui n'est pas optimal. Filtrer les différents canaux et sommer les résultats n'est pas la méthode idéale pour prendre en compte les différents canaux de couleur. En effet, si les directions des gradients sont opposées sur l'ensemble des canaux, la somme des directions s'annule, ce qui amène à la perte des contours ([40]). Pour éviter ce problème, nous utilisons non pas les directions des gradients mais leurs orientations qui sont comprises entre $[0..\pi[$ (à l'inverse des directions qui sont elles comprises entre $[0..2\pi[$). Ces orientations sont données par les tenseurs images ([13]).

Pour une image I donnée, le tenseur local de structure est donné par :

$$\mathbf{G} = \begin{bmatrix} \overline{I_x \cdot I_x} & \overline{I_x \cdot I_y} \\ \overline{I_y \cdot I_x} & \overline{I_y \cdot I_y} \end{bmatrix}, \tag{5.2}$$

où I_x et I_y sont les gradients horizontaux et verticaux de l'image I et l'opérateur $(^-)$ est la convolution par une gaussienne. Ce tenseur est aussi bien utilisé pour la détection de points d'intérêts dans les images que de contours ([146]). De manière équivalente, pour une image $\boldsymbol{I} = (I_1, \ldots, I_n)$ avec plusieurs canaux de couleur, le tenseur est donné par :

$$\mathbf{G} = \begin{bmatrix} \overline{\boldsymbol{I}_x \cdot \boldsymbol{I}_x} & \overline{\boldsymbol{I}_x \cdot \boldsymbol{I}_y} \\ \overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_x} & \overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_y} \end{bmatrix}. \tag{5.3}$$

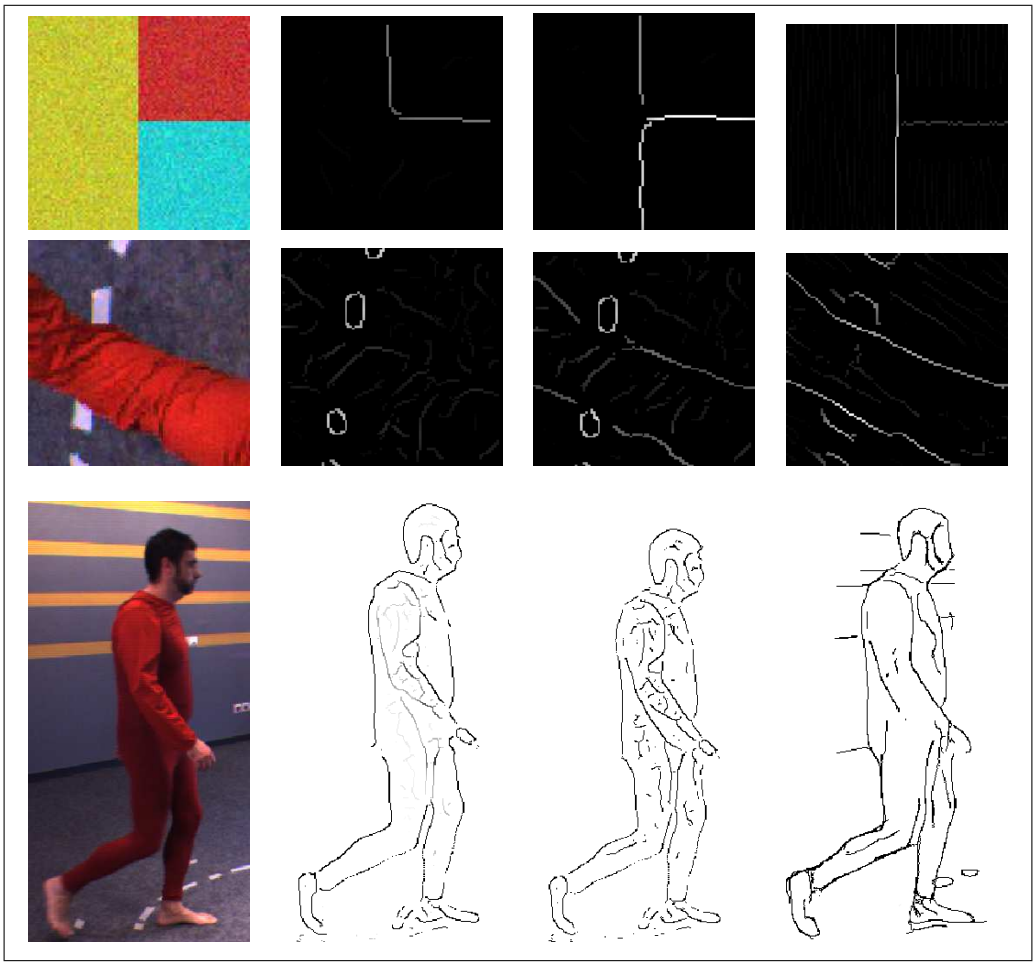


FIG. 5.3: L'extraction des contours orientés permet d'extraire les contours désirés et éviter les contours distrayants pour le suivi de l'acteur. De gauche à droite : L'image couleur ; Les contours extraits à l'aide d'un filtre de Canny standard ; Les contours extraits avec un filtre de Canny sur les images couleurs ; Enfin les contours extraits en utilisant la méthode des contours orientés.

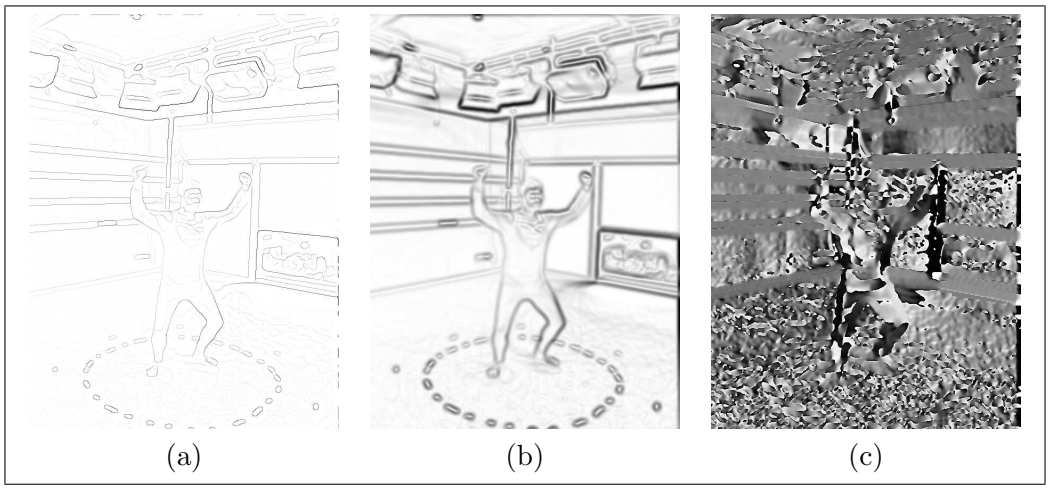
Les valeurs propres du tenseur permettent de déterminer les énergies des dérivées locales dans l'image. Pour le tenseur donné avec l'équation (5.3), les valeurs propres sont données par :

$$\lambda_1 = \frac{1}{2}\left(\overline{\boldsymbol{I}_x \cdot \boldsymbol{I}_x} + \overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_y} + \left((\overline{\boldsymbol{I}_x \cdot \boldsymbol{I}_x} - \overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_y})^2 + 4(\overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_x})\right)^{1/2}\right), \quad (5.4)$$

$$\lambda_2 = \frac{1}{2}\left(\overline{\boldsymbol{I}_x \cdot \boldsymbol{I}_x} + \overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_y} - \left((\overline{\boldsymbol{I}_x \cdot \boldsymbol{I}_x} - \overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_y})^2 + 4(\overline{\boldsymbol{I}_y \cdot \boldsymbol{I}_x})\right)^{1/2}\right). \quad (5.5)$$

Le vecteur propre associé à λ_1 permet de déterminer l'orientation de l'énergie différentielle dominante. Le vecteur propre associé à λ_2 donne l'orientation de l'énergie perpendiculaire à celle prédominante. La somme de ces deux valeurs propres donne l'énergie différentielle totale.

En pratique, nous calculons l'orientation (c.f. figure 5.4-(a)) et la norme (c.f. figure 5.4-(b)) de l'énergie différentielle totale. Puis nous effectuons une suppression des non maxima locaux ([22]). Au final, nous obtenons une carte de gradients correspondant aux contours extraits d'une image couleur (c.f. figure 5.4-(c)).



(a) (b) (c)

FIG. 5.4: La carte des contours (a) est extraite en calculant la norme (b) et la direction (c) des gradients dans les images.

Nous donnons avec la figure 5.5 un exemple de détection de contours. Nous pouvons remarquer un nombre important de contours parasites autres que ceux que nous recherchons. L'ajustement des paramètres de détection aide à éliminer certains des contours parasites soit en faisant varier la taille de la gaussienne pour le filtrage ou les paramètres de seuillage pour l'élimination des faibles gradients. Cependant, pour obtenir des contours idéaux, il faudrait ajuster les paramètres pour chacune des caméras mais aussi pour chacune des images de chaque séquence vidéo. Modifier ces différents paramètres pour chacune des images n'est pas faisable. De plus, l'élimination des contours parasites peut entraîner l'élimination des contours utiles pour la détection.

En outre, les méthodes d'extraction de contours standards détectent les contours dans toute l'image. Nous devons donc séparer les contours liés à l'acteur de ceux liés

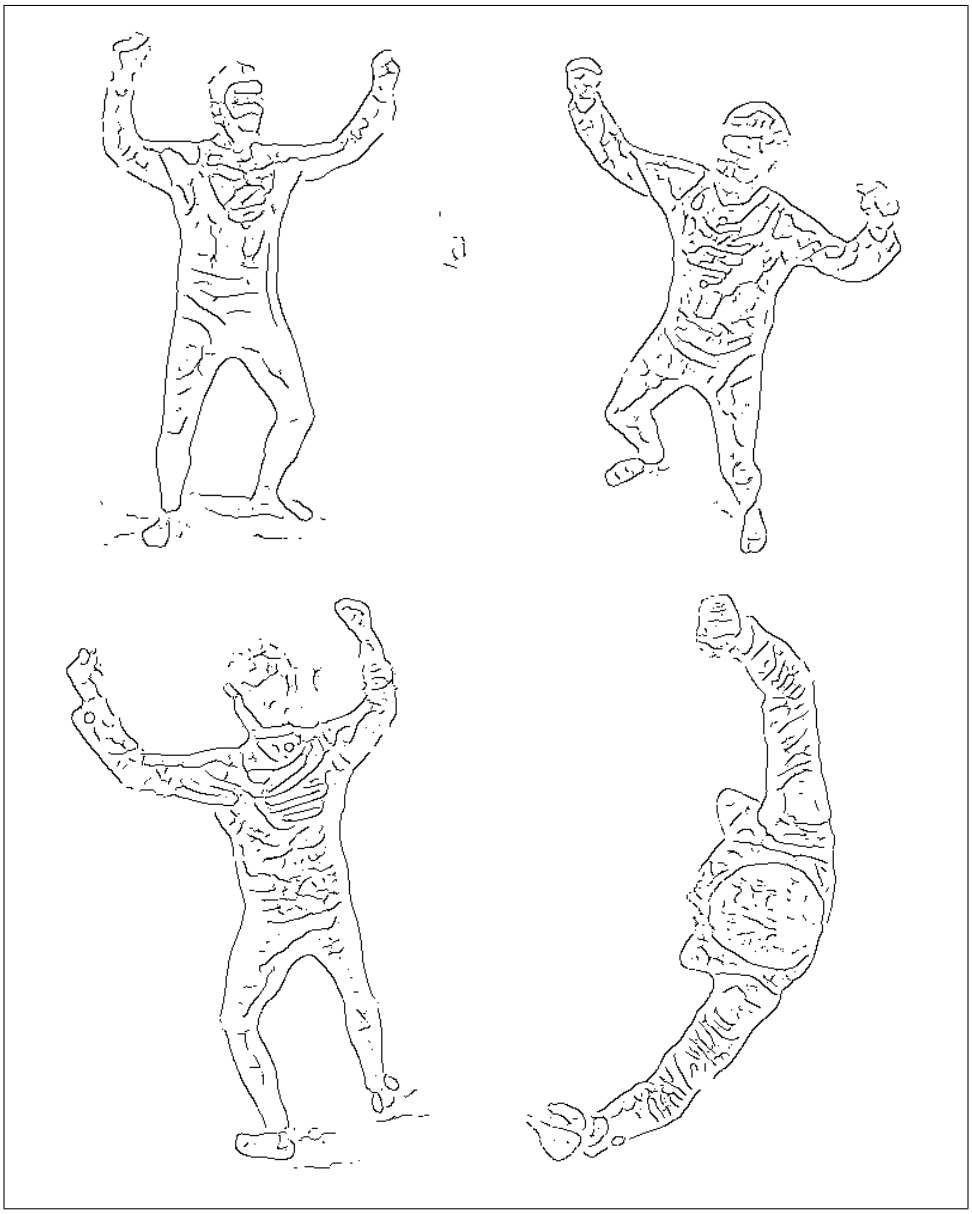


FIG. 5.5: Ces cartes de contours sont extraites en utilisant un filtrage de Canny sur les images couleurs. Ces cartes sont très bruitées. Beaucoup de contours parasites apparaissent.

au reste de l'image. Pour effectuer cette séparation, nous pouvons utiliser la silhouette comme masque. Nous pouvons aussi effectuer un filtrage temporel sur les contours : les contours statiques au cours du temps sont éliminés ([52]).

5.1.2.3 Extraction des contours utilisant le modèle

Nous allons décrire ici la méthode de détection des contours que nous utilisons dans notre algorithme. Nous l'avons construite pour palier au problème de la détection des contours parasites lors de l'utilisation de détecteurs standards. Elle permet d'extraire des images les contours extrémaux de l'acteur tout en minimisant les contours parasites.

Cette méthode s'appuie sur l'hypothèse que nous effectuons un suivi temporel de l'acteur. Nous connaissons donc la pose de l'acteur à l'instant t et nous allons nous aider de cette connaissance pour effectuer la détection des contours à l'instant $t+1$. La pose de l'acteur estimée à l'instant t nous permet de prédire l'orientation des contours dans l'image à l'instant $t+1$. Nous projetons donc dans chacune des images à l'instant $t+1$ le modèle *3D*. Ces contours nous donnent alors une estimation de l'orientation des contours à extraire. Nous utilisons l'orientation donnée par le contour pour construire un filtre permettant de détecter les contours dans la direction prédite. Ce filtrage, que nous allons expliciter dans le suite de ce paragraphe, permet d'améliorer très sensiblement la détection de contours. En effet, il permet d'améliorer la détection des contours dans la direction choisie tandis qu'il supprime les contours parasites (c.f. figure 5.6).

Le filtrage anisotropique Le filtrage utilisé pour effectuer la détection de contours orientés est un filtrage anisotropique gaussien. Les paramètres de la gaussienne anisotropique, son orientation et ses dimensions, sont déduits de la projection du modèle dans l'image. Cela nous permet de cibler la taille et l'orientation des contours à extraire. D'une part l'orientation du contour projeté permet de détecter les contours de l'image orientés de la même manière et d'autre part la taille de la gaussienne peut être choisie de sorte que les contours recherchés soient détectés au mieux. Prenons l'exemple de la cuisse. Une gaussienne de petite taille permettra de détecter les contours associés à la cuisse mais aussi des contours parasites. Si nous prenons une gaussienne dont la valeur propre principale est de dimension comparable à celle de la longueur de la cuisse, les contours de la cuisse seront détectés contrairement aux les contours parasites.

Le noyau du filtre gaussien anisotropique ([89], [54]) que nous utilisons a pour expression :

$$g(u,v,\theta;\sigma_u,\sigma_v) = \frac{1}{2\pi\sigma_u\sigma_v} e^{-(\frac{u^2}{2\sigma_u^2}+\frac{v^2}{2\sigma_v^2})}, \tag{5.6}$$

où $(u,v)^\top = \mathbf{r}(\theta)\boldsymbol{x}$ avec $\mathbf{r}$ la matrice de rotation 2×2 d'angle θ donné par l'orientation du contour extrémal dans l'image. $\sigma_u = length/4$ est donné par la longueur du contour extrémal dans l'image. Nous choisissons $\sigma_v = 2$ de manière expérimentale avec comme condition $\sigma_u > \sigma_v$. Le choix de σ_v permet d'avoir une

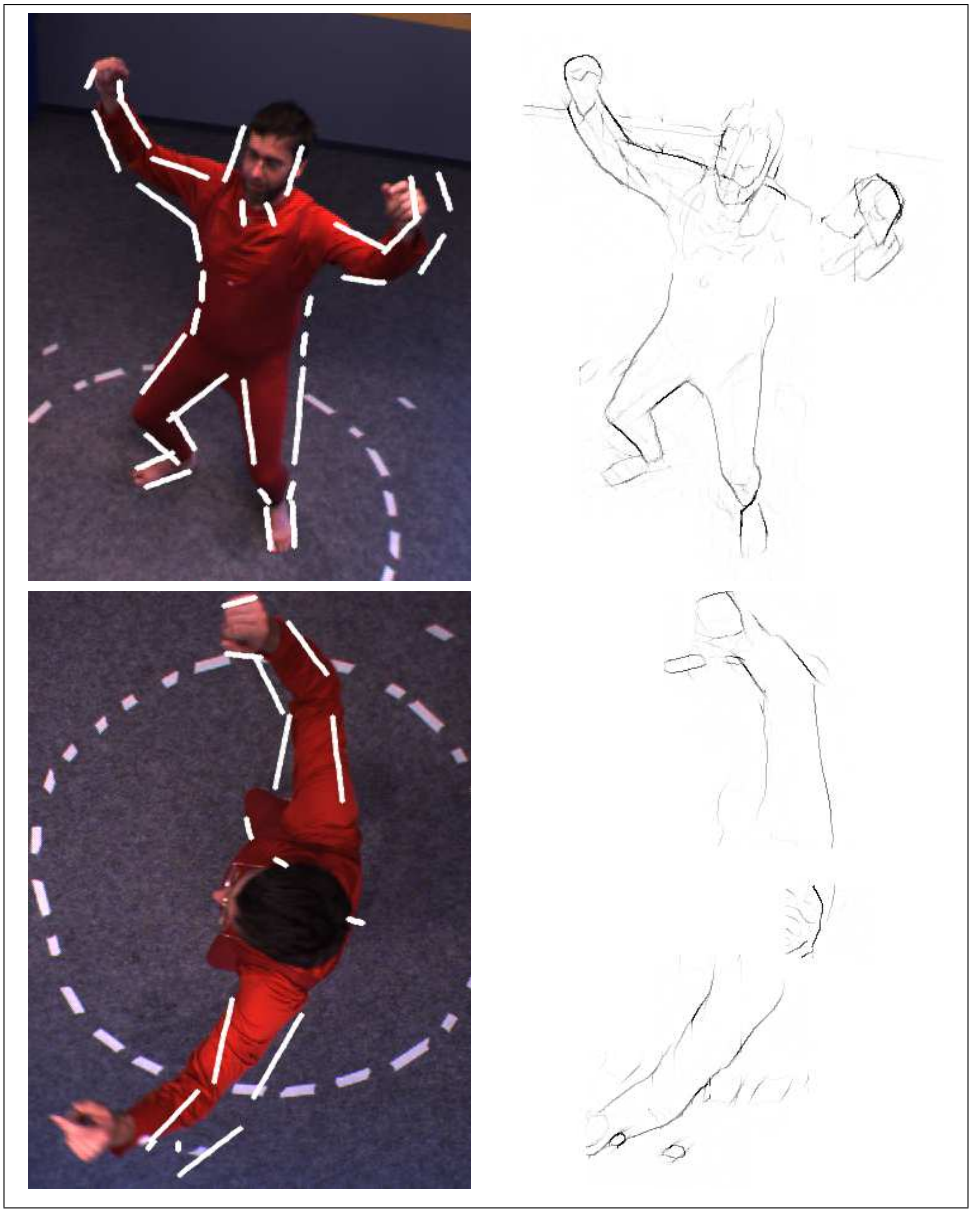


FIG. 5.6: Les gradients sont calculés en utilisant un filtrage orienté. Les contours sont donc renforcés pour la direction recherchée et affaiblis pour des directions perpendiculaires. Dans cette illustration, les images de gradient sont en fait une superposition de l'ensemble des contours détectés pour l'ensemble des parties du corps.

certaine tolérance dans la recherche du contour autour de l'orientation prédite (prendre une valeur plus petite réduirait cette tolérance).

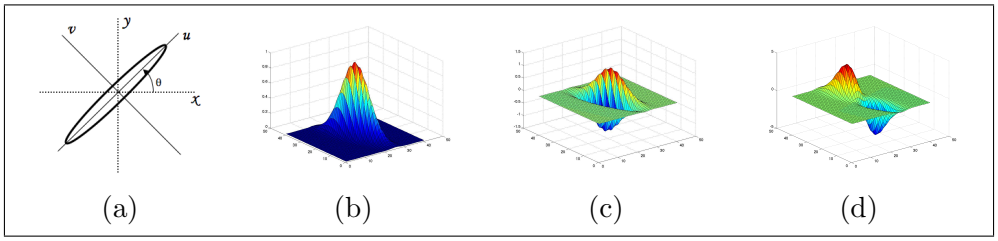


FIG. 5.7: Pour effectuer la détection de contours orientée, nous utilisons l'orientation ainsi que les dimensions des contours projetés (a) et (b). Les dérivées selon les axes de la gaussienne sont données sur les figure (c) et (d).

Pour effectuer la détection, nous utilisons les implémentations proposées dans [152] et [144]. Ce sont des implémentations efficaces et rapides du filtrage anisotropique gaussien.

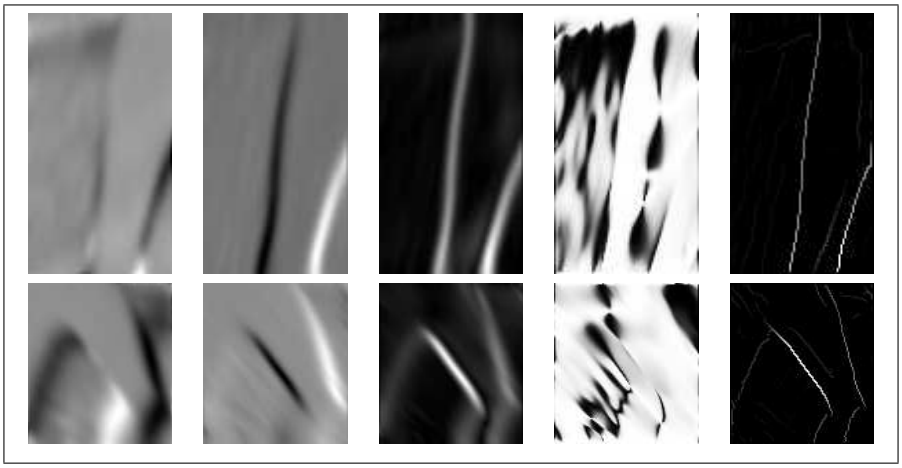


FIG. 5.8: Après avoir calculé le gradient selon u et v, nous calculons la norme et l'orientation des gradients dans les *patchs*. En utilisant ces deux dernières images, nous déduisons les contours dans les *patchs*.

La détection des contours pour une direction donnée n'est utile que dans une région proche du contour modèle prédit. Nous effectuons donc une détection locale des contours : pour chacun des contours, nous déterminons une région (que nous nommerons *patch*) dans laquelle nous effectuons la détection de contour orientée. Nous illustrons les différentes étapes du calcul du gradient avec la figure 5.8. La dernière étape est en fait une suppression des non-maxima locaux.

La figure 5.6 est une illustration de la détection de contours pour l'ensemble de l'acteur. Les contours extraits sont nettement meilleurs que ceux obtenus en utilisant un détecteur standard. Cependant, il reste encore des contours parasites que nous pouvons éliminer en seuillant les gradients. En pratique, nous extrayons les

contours en effectuant un chaînage par hystérésis des cartes de gradients. Nous obtenons alors des cartes de contours binaires tel que nous l'avons illustré sur la figure 5.9.

Au final, nous obtenons un détecteur de contours se rapprochant d'un détecteur de contours extrémaux idéal.

5.1.3 La couleur

Les contours apparents des cônes sont échantillonnés. En chacun des points échantillonné la normale au contour est calculée. La couleur moyenne est calculée le long de ces normales à l'intérieur et à l'extérieur du cône (c.f. figure 5.10). Le nombre de pixels sur lesquels la valeur moyenne est calculée dépend de la proximité des contours recherchés. Plus les contours sont loin plus le nombre de pixels est élevé. L'objectif est alors de minimiser la variance de la couleur sur chacune des moitiés de la normale. Nous verrons l'expression analytique de cette erreur au paragraphe 5.2.1. La valeur de cette erreur dépend des paramètres du modèle *3D*. La minimisation de cette erreur permet aussi bien de dimensionner le modèle *3D* que d'estimer la pose de l'acteur.

Cependant, la méthode de recherche des contours couleur est pénalisée dans le cas où la différence entre les couleurs de deux parties du corps ou entre le corps et le fond de l'image n'est pas suffisamment grande. Nous montrerons quelques résultats utilisant cette technique dans le chapitre 6.

5.1.4 Discussions

Nous avons vu que nous pouvions utiliser les silhouettes, les contours ou encore la couleur pour effectuer le suivi du mouvement dans les images.

Nous avons vu que l'utilisation des silhouettes est restreinte au cadre d'environnements d'acquisition contraints (lumière statique, fond de l'image statique, etc.). La détection de contours standard provoque l'apparition de contours bruités et parasites. Pour palier à cette dernière difficulté, nous avons présenté une méthode de détection de contours basé modèle. Cette méthode permet d'obtenir des contours sensiblement meilleurs. D'une part, ils correspondent mieux à l'observation et d'autre part il y a moins de contours parasites.

Nous devons tout de même noter des limitations à l'utilisation de cette détection orienté modèle. Nous allons les aborder ici tout en montrant que nous pouvons les résoudre ou tout du moins les contourner.

Les contours francs du fond de l'image Nous effectuons une détection de contours orientés. Cependant, si les contours de la scène sont francs, ceux-ci sont détectés même s'ils sont orthogonaux à la direction recherchée. Ce problème est lié à la largeur de la gaussienne utilisée pour le filtrage. Nous pouvons éliminer les derniers contours distrayants en comparant l'orientation du modèle à l'orientation

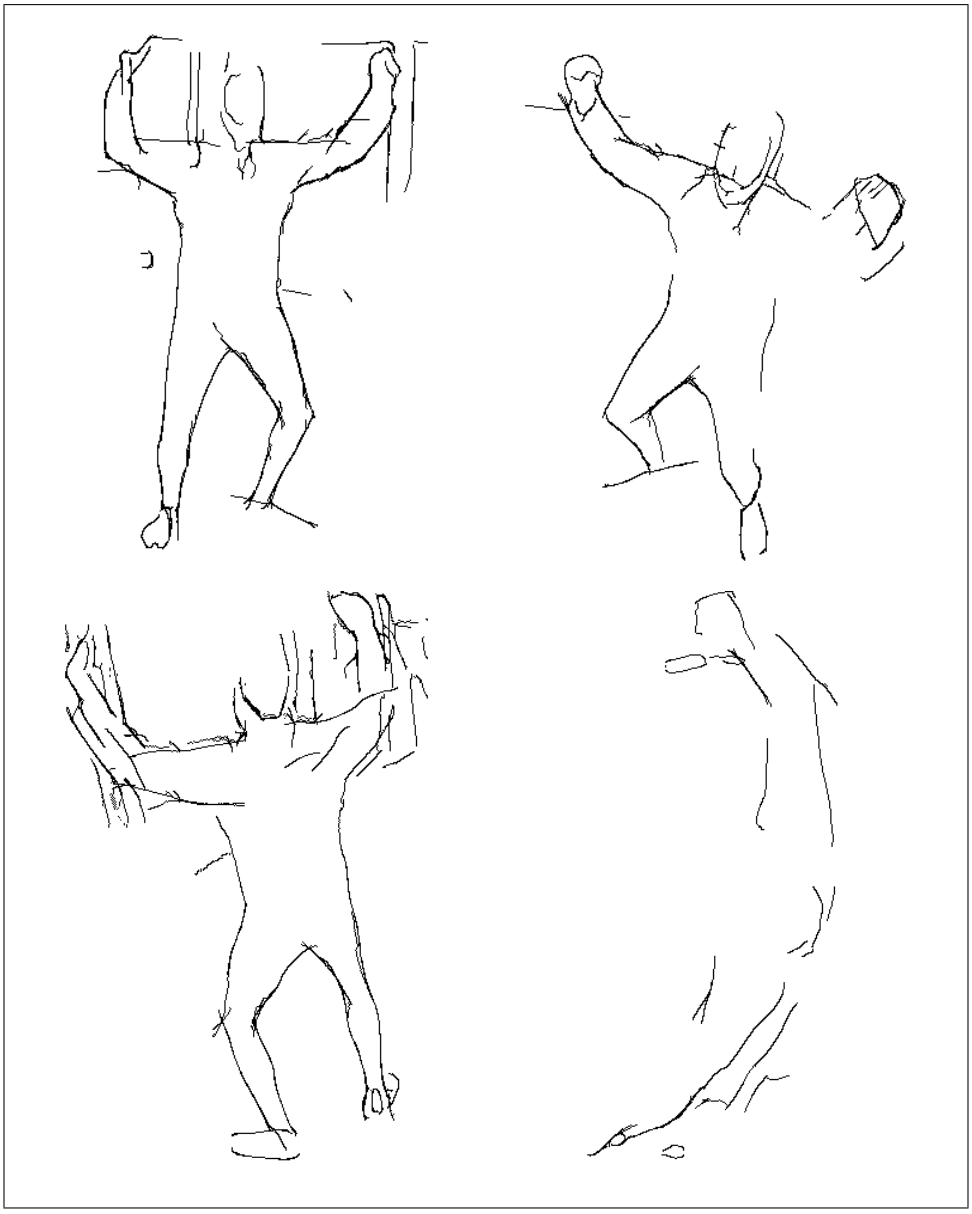


FIG. 5.9: Une fois le gradient orienté calculé pour chacun des contours modèle, les contours sont extraits en utilisant une méthode de chaînage. L'illustration présente une superposition de l'ensemble des contours extraits dans chacun des *patchs*.

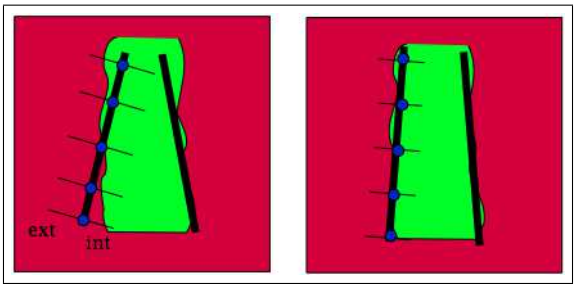


FIG. 5.10: Les paramètres des cônes sont estimés en maximisant l'écart entre la moyenne à l'extérieur et à l'intérieur du cône, et ce le long de la normale au contour projeté dans l'image.

des contours. Si l'orientation des contours observés n'est pas proche de celle du modèle, alors les contours sont éliminés.

Méthode locale La détection des contours se fait localement et non dans toute l'image, ce qui empêche le suivi de mouvements amples ou rapides. Pour palier à ce problème, nous combinons l'approche utilisant les silhouettes avec celle utilisant les contours orientés. Plus précisément, la méthode utilisant les contours orientés permet d'affiner l'estimation de la pose effectuée avec la méthode de détection standard des contours.

Nous avons aussi abordé l'utilisation de la couleur pour effectuer le suivi. Contrairement à l'utilisation des contours, nous allons voir dans le prochain paragraphe que l'utilisation de la couleur ne permet pas une mise en correspondance aussi aisée. Les calculs sont plus lourds que pour les contours.

5.2 Mise en correspondance

L'estimation des paramètres de pose du modèle ou le dimensionnement de celui-ci nécessite de mettre en place une mesure de l'écart entre les observations et la prédiction donnée par le modèle *3D*.

Nous allons aborder dans cette partie le problème de la mise en correspondance des données observées avec les données du modèle. Nous expliciterons alors le calcul de l'erreur entre le modèle et les observations pour chacune des méthodes que nous avons présentées dans la partie précédente.

5.2.1 Utilisation de la couleur

Nous allons expliciter dans ce paragraphe la méthode d'estimation de l'erreur utilisant la couleur pour estimer les paramètres du modèle *3D*.

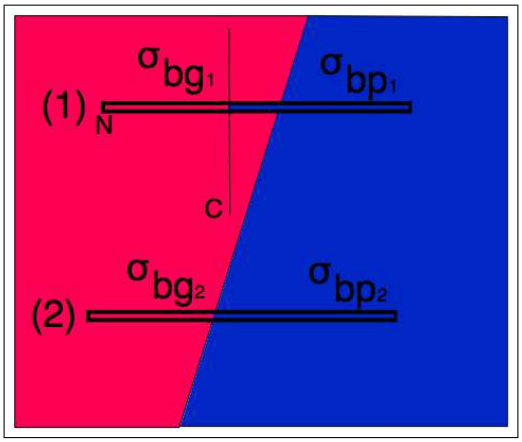


FIG. 5.11: Illustration du calcul de la variance sur la couleur. C est le contour du modèle projeté dans l'image et N la normale au contour en un point donné. Nous avons $\sigma_{bg_1} > \sigma_{bg_2}$. De même, $\sigma_{bp_1} > \sigma_{bp_2}$. La position où la variance est minimale est la position 2.

La fonction de coût Considérons une image avec deux couleurs (rouge et bleu) séparées par une frontière bien définie (c.f. figure 5.11). Nous voulons trouver l'équation de la droite qui correspond au mieux à cette frontière. Pour cela, nous échantillonnons une droite et nous calculons les normales à cette droite en chacun de ces points. Chacune des normales est à nouveau échantillonnée. En chacun des points (de la normale) la valeur de la couleur est lue. Nous calculons alors la différence entre la valeur lue et la valeur moyenne attendue (rouge à gauche de la droite et bleu à droite, par exemple). L'objectif est de minimiser l'erreur entre la valeur lue et la valeur moyenne en modifiant les paramètres de la droite.

En pratique, il s'agit de calculer la variance de la couleur à gauche et à droite du contour projeté. Pour chacun des points sur chacune des normales, nous avons :

$$\sigma_g^2(\boldsymbol{x}) = (r(\boldsymbol{x}) - \mu_r)^2 + (g(\boldsymbol{x}) - \mu_{g_g})^2 + (b(\boldsymbol{x}) - \mu_{b_g})^2, \quad (5.7)$$

$$\sigma_d^2(\boldsymbol{x}) = (r(\boldsymbol{x}) - \mu_{r_g})^2 + (g(\boldsymbol{x}) - \mu_{g_g})^2 + (b(\boldsymbol{x}) - \mu_{b_g})^2, \quad (5.8)$$

où r, g, b sont les canaux de couleur (rouge, vert, bleu). $\boldsymbol{x}$ est un point de la normale. Les indices $_d$ et $_g$ dénotent les canaux de couleur à gauche et à droite des contours. μ est la valeur moyenne. σ_g^2 est la variance de la couleur à gauche du contour. σ_d^2 est la variance de la couleur à droite du contour. Si le modèle du fond ou de l'acteur sont connus alors les moyennes calculées pour déterminer la variance sont celles des modèles. Dans le cas contraire, les moyennes sont évaluées le long de chaque normale.

L'objectif est alors de minimiser la fonction d'erreur suivante :

$$E(\mathcal{X}, \mathcal{Y})_{color} = \sum_{b=0}^{N_{bp}} \sum_{i=0}^{2N} \sum_{j=0}^{N_n} (\sigma_g^2(\boldsymbol{x}_i + j\boldsymbol{\delta x}) + \sigma_d^2(\boldsymbol{x} - j\boldsymbol{\delta x})). \quad (5.9)$$

La notation $j\boldsymbol{\delta x}$ dénote le j^{e} incrément le long de la normale au contour. $\boldsymbol{\delta x}$ est le vecteur directeur de la normale. N_{bp} est le nombre de partie du corps dans le modèle *3D*, N le nombre de point le long de chaque contour et N_n le nombre de points sur chacune des normales.

La lecture de la valeur de la couleur dans l'image ne peu se faire qu'en utilisant une interpolation que nous avons choisie bilinéaire. $r_{\mathcal{Y}}(\boldsymbol{x})$ s'écrit donc de la manière suivante :

$$\begin{aligned} r_{\mathcal{Y}}(\boldsymbol{x}) &= \alpha\beta C_{\mathcal{Y}}(u+1, v+1) + (1-\alpha)\beta C_{\mathcal{Y}}(u, v+1) + \\ &\quad \alpha(1-\beta) C_{\mathcal{Y}}(u+1, v) + (1-\alpha)(1-\beta) C_{\mathcal{Y}}(u, v), \end{aligned} \tag{5.10}$$

où, si $\boldsymbol{x} = (x_1, x_2)$ et si [ ] représente la fonction partie entière, alors $u = [x_1]$, $v = [x_2]$, $\alpha = x_1 - u$, $\beta = x_2 - v$. Chacun de ces paramètres dépend des paramètres du modèle *3D*. Enfin, $C_{\mathcal{Y}}$ est le canal de couleur et $C_{\mathcal{Y}}(u, v)$ représente la valeur lue au pixel (u, v).

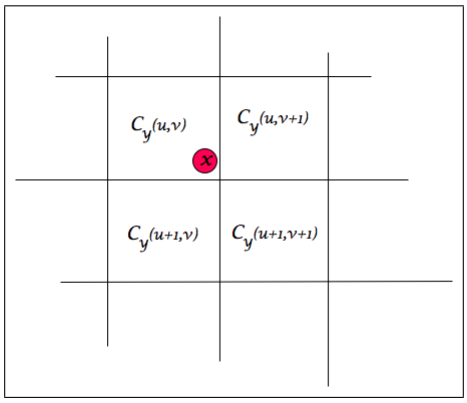


FIG. 5.12: Illustration de l'interpolation bilinéaire.

La fonction d'erreur dépend donc directement des paramètres de pose de l'acteur. De plus, l'interpolation choisie est une fonction continue et dérivable donc $E(\mathcal{X}, \mathcal{Y})_{color}$ l'est aussi. Pour minimiser cette erreur, nous pourrons utiliser des techniques de minimisation non linéaire mais continues.

5.2.2 Utilisation des contours

Pour estimer la pose de l'acteur lors du suivi du mouvement, nous disposons des silhouettes de l'acteur, de la carte des contours de l'acteur et des contours projetés du modèle. Il s'agit donc de mesurer un écart entre deux ensembles de points (ceux du contour image et ceux du contour modèle). Or pour effectuer la comparaison entre deux ensembles de points, peu de possibilités existent. Nous pouvons citer des distances de type distance de Fréchet ([5]), Earth Mover's Distance ([60]) ou encore la distance de Hausdorff ([76]). Nous avons décidé d'utiliser la distance de Hausdorff comme mesure de l'erreur. Généralement considérée comme non-dérivable, nous l'adaptons pour créer une fonction de coût continue et dérivable.

5.2.2.1 La distance de Hausdorff

Nous considérons généralement une distance comme étant la longueur du plus court chemin entre deux points. Si nous considérons la distance entre un point et un polygone ce sera le plus court chemin du point à l'arête (du polygone) la plus proche. Dès qu'il s'agit de la distance entre deux ensembles de points, la notion de distance comme plus court chemin peut devenir moins satisfaisante. Si nous considérons la figure 5.2.2.1, la distance entre les sommets dessinés en plein est-elle une mesure de la distance entre les deux polygones ? Nous sommes plutôt satisfait d'une distance qui est petite si tous les sommets d'un polygone sont proches des sommets de l'autre polygone. La notion de chemin minimum ne peut donc plus être vérifiée dans ce cas. La distance de Hausdorff donne une nouvelle formulation de la distance et est appropriée au cas des ensembles de points.

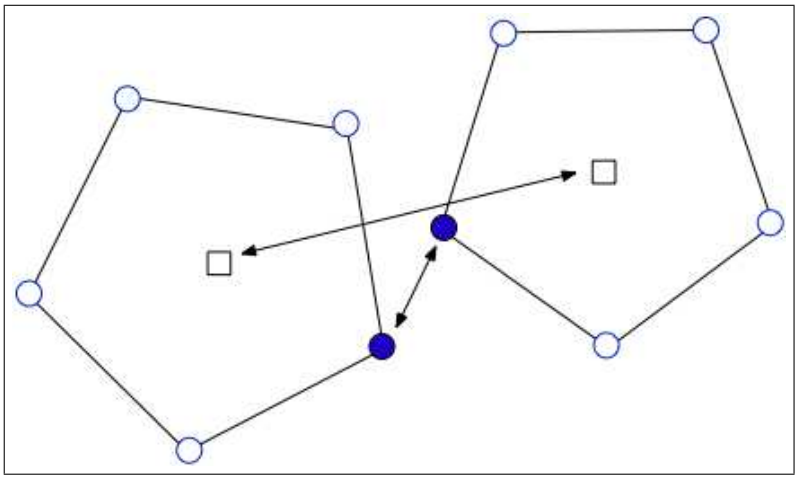


FIG. 5.13: La distance entre les deux polygones peut être définie de différentes manières. La distance entre les sommets dessinés en plein, entre les centres des polygones, etc.

De manière formelle, la distance de Hausdorff entre deux ensembles de points $\mathcal{X}$ et $\mathcal{Y}$ est donnée par :

$$H(\mathcal{X}, \mathcal{Y}) = \max(h(\mathcal{X}, \mathcal{Y}), h(\mathcal{Y}, \mathcal{X})), \tag{5.11}$$

où $h(\mathcal{X},\mathcal{Y})$ est définit par :

$$h(\mathcal{X}, \mathcal{Y}) = \max_{x \in \mathcal{X}} \left\{ \min_{y \in \mathcal{Y}} \{d(x, y)\} \right\}, \tag{5.12}$$

où x et y sont des points appartenant aux ensembles de points $\mathcal{X}$ et $\mathcal{Y}$ et où $d(x, y)$ est une métrique quelconque entre ces points.

H définit la distance entre $\mathcal{X}$ et $\mathcal{Y}$. L'équation (5.12) définit une distance de Hausdorff de $\mathcal{X}$ vers $\mathcal{Y}$ et est appelée la distance de Hausdorff dirigée (*directed Hausdorff distance*). $h(\mathcal{X}, \mathcal{Y})$ et $h(\mathcal{Y}, \mathcal{X})$ peuvent être aussi appelées distances de Hausdorff *foward*

et *backward*. Notons que h ne définit pas une distance au sens mathématique, puisque la propriété de symétrie n'est pas vérifiée : $h(\mathcal{X}, \mathcal{Y}) \neq h(\mathcal{Y}, \mathcal{X})$.

Nous rappelons que $\mathcal{Y}$ est l'ensemble des points des contours observés et $\mathcal{X}$ l'ensemble des points appartenant aux contours apparents du modèle. Il s'agit donc de calculer la distance de l'ensemble des points du contour observé à l'ensemble des points du contour projeté et vice et versa.

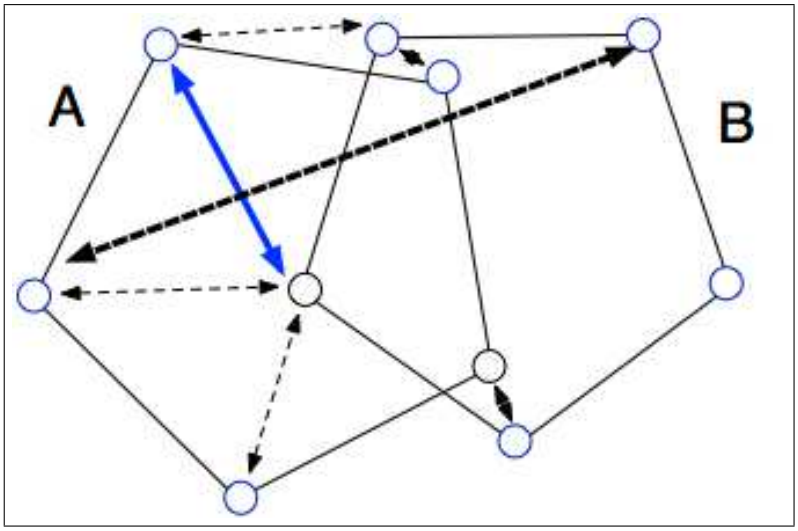


FIG. 5.14: La distance de Hausdorff orientée de A vers B (flèche en gras plein) est différente de la distance de B vers A (flèche en gras pointillée). La distance de A vers B est la plus grande des distances des points de A à l'ensemble B (flèches en pointillés)

La distance de Hausdorff s'appuie sur une métrique d permettant de calculer la distance entre deux points. Le choix le plus courant est la distance euclidienne. Cette métrique est adaptée à des ensembles de points de faible cardinal. Il s'agit en effet de calculer la distance euclidienne point à point pour l'ensemble des points des deux espaces considérés. Pour des ensembles de points importants, le calcul de la distance euclidienne peut devenir coûteuse en terme de temps de calcul. De plus, il faut déterminer le minimum des distances, ce qui nécessite de chercher pour un point du contour observé (resp. projeté) le point le plus proche du contour projeté (resp. observé). Il s'agit aussi d'une opération qui peut s'avérer coûteuse.

Enfin, la fonction max n'est pas une fonction continue. L'utilisation de la distance de Hausdorff telle quelle n'est donc pas adéquate dans un cadre de l'optimisation non linéaire.

Nous allons dans un premier temps modifier la distance pour la rendre continue et dérivable. Ensuite pour éviter le calcul de la fonction min pour tous les points, nous introduirons la transformée en distance comme métrique. Cette transformée permet de calculer rapidement les distances et le min. Enfin, alors que beaucoup d'approches utilisent une distance de Hausdorff orientée, nous allons montrer que, de par le choix

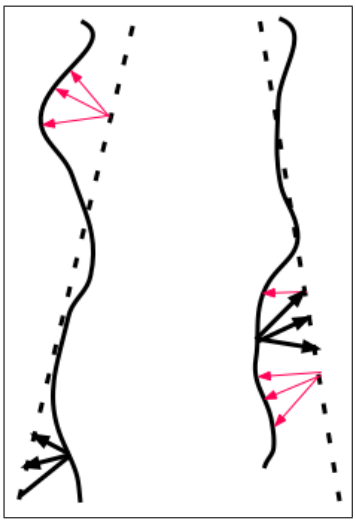


FIG. 5.15: Lorsque le nombre de points de chacun des ensembles est élevé, la recherche du minimum pour le calcul de la distance de Hausdorff peut s'avérer coûteuse.

de notre modèle, nous pouvons conserver la distance de Hausdorff avec son expression symétrique.

Modification de la distance de Hausdorff : La fonction max n'est pas une fonction continue et n'est donc pas dérivable. Pour palier à ce problème, la solution généralement retenue (et que nous retenons aussi) est de remplacer la fonction max de la distance orientée et de la distance de Hausdorff par une somme sur l'ensemble des termes.

La nouvelle expression de la distance orientée est alors :

$$h(\mathcal{X},\mathcal{Y}) = \sum_{x\in\mathcal{X}} \left(\min_{y\in\mathcal{Y}} \{d(x,y)\} \right). \tag{5.13}$$

La métrique : Pour calculer rapidement le min, nous utilisons la transformée en distance permettant de déterminer la distance d'un point quelconque de l'espace à un point d'un ensemble donné. D'une part, pour calculer $h(\mathcal{X},\mathcal{Y})$, nous effectuons la transformée en distance sur la carte des contours. D'autre part, pour calculer $h(\mathcal{Y},\mathcal{X})$, nous calculons la transformée en distance sur les contours du modèle. Cependant, lors de la minimisation, les contours prédits du modèle *3D* évoluent au cours des itérations. Cette seconde transformée doit donc être réévaluée à chaque itération de la minimisation. Le calcul de la transformée n'est pas très coûteux mais la ré-évalutation de la carte à toute les itérations peut surcharger le processeur et ralentir les calculs. Peu de travaux utilisent donc cette seconde moitié de la distance de Hausdorff. Nous verrons que le choix du modèle *3D* nous permet de calculer rapidement cette carte de distance, ou tout du moins une approximation de celle-ci, et que nous pouvons donc garder la distance de Hausdorff complète.

Nous abordons maintenant le calcul de $h(\mathcal{X}, \mathcal{Y})$, qui est la distance Image-Modèle. Puis nous aborderons le calcul de $h(\mathcal{Y}, \mathcal{X})$, qui est la distance Modèle-Image dans la section suivante.

5.2.2.2 La distance Image-Modèle

Il existe plusieurs méthodes pour calculer la transformée en distance sur des cartes binaires ([32] ou [15]). Le résultat de la transformée en distance fournit une carte des distances. Chaque pixel de la carte nous donne la distance au point de contour le plus proche.

Nous pouvons calculer la carte des distances à partir des silhouettes ou encore des contours (figure 5.16). Dans le cas de la détection orientée modèle, nous calculons la transformée en distance au voisinage des contours détectés (figure 5.17).

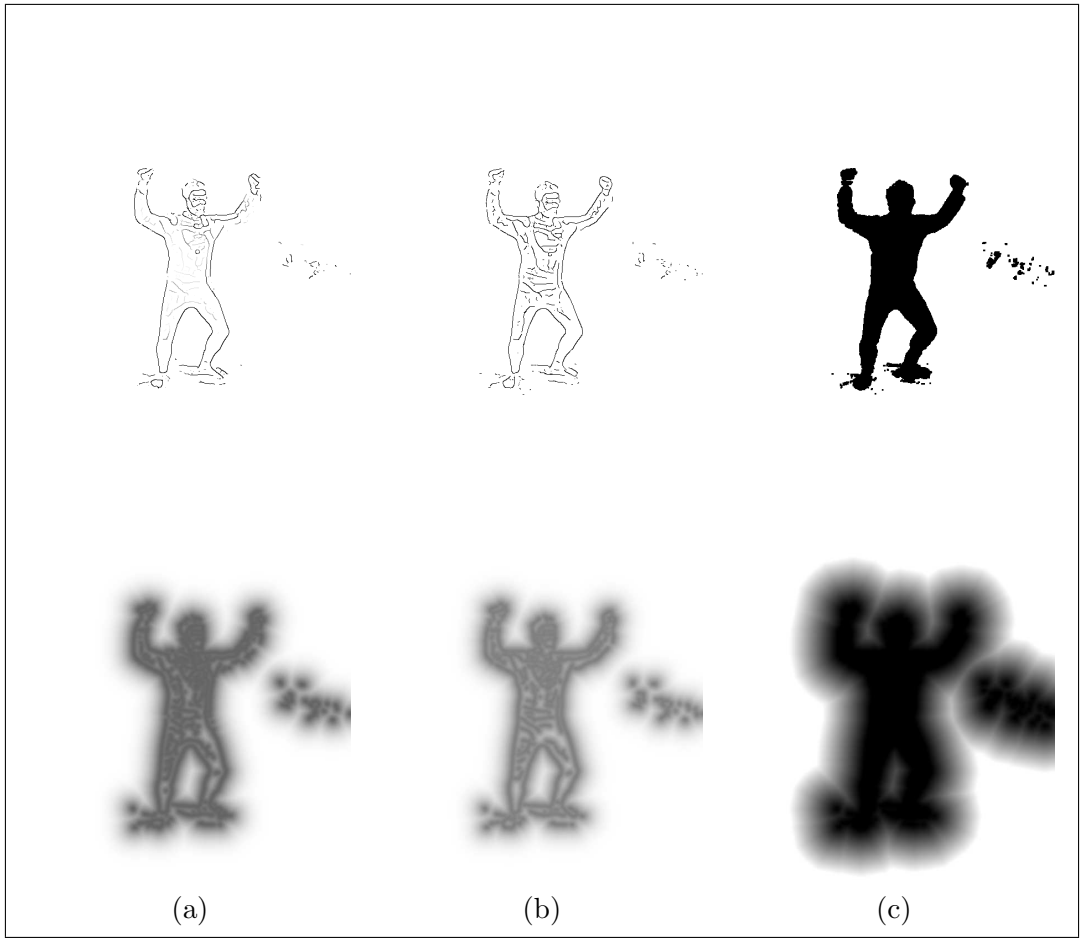


FIG. 5.16: La carte des distances est calculée à partir d'une image binaire. Nous pouvons calculer celle-ci sur les contours extraits à partir d'un filtre Canny standard (a), extraits avec un filtrage sur la couleur (b) et enfin sur les silhouettes (c).

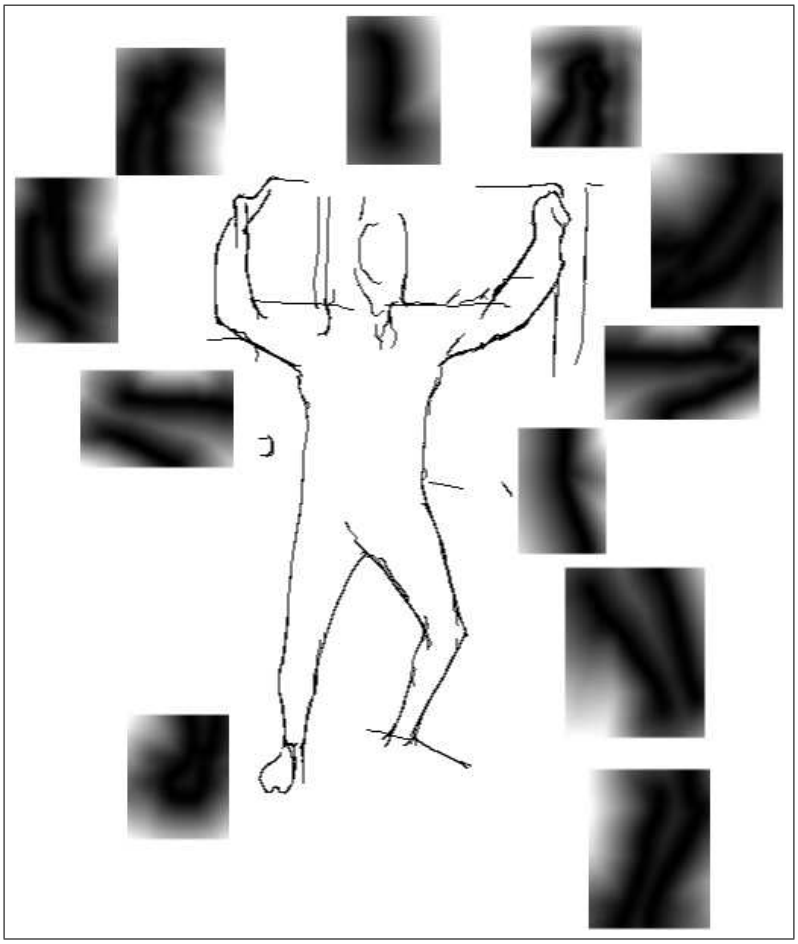


FIG. 5.17: La carte des distances est calculée sur chacun des *patchs* utilisés pour effectuer la détection de contours orientée.

Remarque : Si la méthode d'extraction des contours utilisée est celle de Canny, il n'y a pas besoin de ré-estimer la carte des distances lors de l'estimation des paramètres (c'est-à-dire pendant la phase de minimisation). En effet, les contours observés dans les images sont statiques au cours de la minimisation. Cependant, dans le cas de l'extraction des contours à l'aide du détecteur basé modèle, nous pouvons choisir de ré-estimer ou non la carte des distances. En effet, la pose du modèle varie au cours du temps pour se rapprocher de la pose correcte de l'acteur. Nous pouvons donc ré-extraire les contours à chaque itération. Cette nouvelle extraction nécessite de recalculer la carte des distances. En pratique, les mouvements que nous pouvons estimer avec cette méthode sont de faible amplitude. L'extraction des contours se fait donc autour des contours corrects. Par expérience, les contours ne sont pas mieux extraits au cours des itérations. Nous n'avons donc pas besoin d'extraire les contours à chaque itération et les cartes ne sont donc pas ré-estimées.

Estimation de la distance : Nous avons calculé la carte des distances pour les silhouettes et les contours extraits (quelle que soit la technique employée). Nous pouvons maintenant estimer la distance (orientée) des contours modèles aux contours images. Comme nous l'avons vu dans le chapitre 4, nous échantillonnons les contours occultants et projetons dans les images les points échantillonnés. Pour évaluer la distance d'un point du contour extrémal au point de contour observé le plus proche, il suffit donc de lire la valeur de la distance sur la carte des distances.

L'erreur entre les contours projetés par rapport aux contours extraits pour le modèle *3D* complet est donc de la forme :

$$\boxed{h(\mathcal{X},\mathcal{Y}) = \sum_{c=0}^{N_c}\sum_{b=0}^{N_{bp}}\sum_{i=0}^{2N}\delta_{i,c}^{b} D^2_{\mathcal{Y}_c}(\boldsymbol{x}_{i,c}^{b})}, \tag{5.14}$$

où :

- la première somme porte sur les images. En effet, nous utilisons plusieurs caméras et donc l'erreur totale est la somme sur l'ensemble des images à un instant donné des erreurs entre la projection et l'observation.
- la seconde somme porte sur les parties du corps et N_{bp} est le nombre de cônes dans le modèle,
- la troisième somme porte sur les points échantillonnés du contour extrémal, et N est donc le nombre d'échantillons par contour occultant,
- $\delta_{i,c}^{b}$ vaut 1 si le point i du cône b est visible dans l'image c (c.f. 4.3.3), 0 sinon,
- enfin, $D_{\mathcal{Y}_c}(\boldsymbol{x}_{i,c}^{b})$ est la valeur de la distance lue sur la carte de la transformée en distance au point de coordonnées $\boldsymbol{x}_{i,c}^{b}$.

$\boldsymbol{x}_{i,c}^{b}$ est le vecteur de coordonnées d'un point appartenant au contour extrémal et n'est donc pas un vecteur à valeur entière. La lecture de la distance $D_{\mathcal{Y}_c}(\boldsymbol{x}_{i}^{b})$ nécessite donc d'effectuer une interpolation que nous choisissons bilinéaire.

La forme analytique de $D_{\mathcal{Y}}(\boldsymbol{x})$ est donnée par l'équation (5.10), où $C_{\mathcal{Y}}$ devient la carte des distances.

5.2.2.3 La distance Modèle-Image

Nous allons maintenant aborder le calcul de la distance des points du contour image aux points du contour observé.

De la même manière que précédemment, nous aimerions calculer une transformée en distance sur les contours du modèle. Cependant, les contours évoluent très rapidement au cours de l'estimation des paramètres. Nous ne pouvons donc pas nous permettre de calculer la transformée en distance pour toute l'image à chaque modification des paramètres. Nous allons adapter le calcul de la distance, pour pouvoir mettre à jour rapidement la distance.

Pour un point donné du contour extrait de l'image, nous devons calculer la distance au contour projeté le plus proche. Il faut donc dans un premier temps déterminer le contour le plus proche puis calculer la distance. Nous avons vu que la distance de chanfrein permettait d'obtenir rapidement cette distance. Cependant, du fait que les contours modèles évoluent, il est nécessaire de recalculer la carte de chanfrein à chaque itération. Cette opération s'avère coûteuse pour le processeur et ralentit énormément l'estimation de la pose. De plus, lors de l'estimation de la distance Image-Modèle, nous avons utilisé l'interpolation bilinéaire pour rendre la fonction de coût continue et dérivable. Dans le cas présent, l'interpolation n'a plus lieu d'être a priori. En effet, un point de contour est un pixel dont les coordonnées sont à valeur entière donc la lecture de la distance sur la carte de chanfrein engendrée par le modèle ne nécessite plus d'interpolation. La fonction d'erreur n'est donc pas continue (puisque la grille est discrète). Ce désavantage nous a amené à considérer le calcul de la distance d'une manière différente.

Le choix du contour modèle le plus proche d'un point du contour image est effectué à l'aide d'un diagramme de Voronoï. Une fois le contour modèle choisi, nous calculons la distance du point image au segment du contour de manière analytique.

Construction du diagramme de Voronoï

Définition 1 *Soit S un ensemble de n sites de l'espace euclidien en dimension d. Pour chaque site p de S, la cellule de Voronoï V(p) de p est l'ensemble des points de l'espace qui sont plus proches de p que de tous les autres sites de S. Le diagramme de Voronoï de V(S) est la décomposition de l'espace formé par les cellules de Voronoï des sites.*

Notre objectif est de déterminer les cellules de Voronoï et non de connaître de manière analytique les frontières entre les cellules. Une méthode graphique et donc rapide du calcul des cellules suffit dans notre cas.

Dans un plan, nous dessinons les contours projetés du modèle *3D*. Pour chacun de ces contours, nous dessinons deux plans de pente 1 et -1 de sorte qu'ils intersectent le segment (c.f. illustration 5.18-(a)). Ces plans sont en théorie prolongés à l'infini. Pour chacun des segments et donc pour chaque paire de plans, nous associons un identifiant (une couleur). La projection de ces plans dans le plan contenant les contours projetés

en tenant compte des occultations permet de construire les cellules de Voronoï associes à chaque segment (c.f. illustration 5.18-(b)). La carte graphique gère de manière implicite les occultations, ce qui nous a amené à utiliser ses capacités pour calculer les diagrammes.

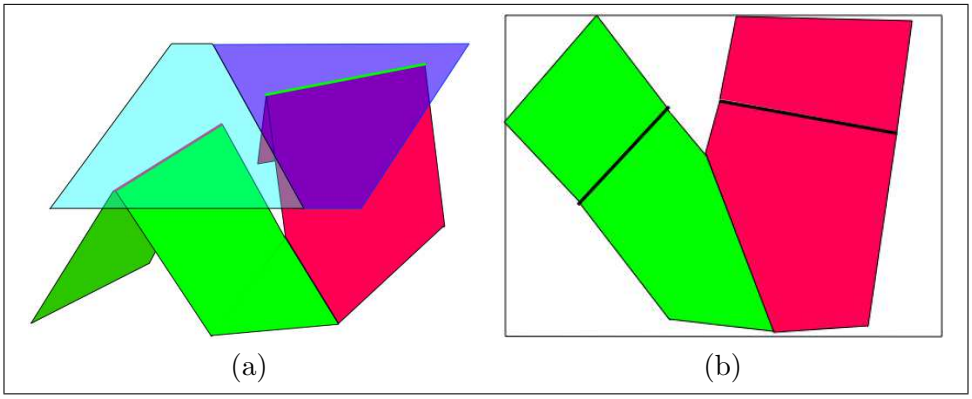


FIG. 5.18: Les régions de Voronoï sont calculées en dessinant des plans de pente 1 et -1 intersectant les segments de droites. L'intersection de ces plans définit l'interface entre deux régions de Voronoï. Nous illustrons le dessin en *3D* (a). Nous illustrons la projection dans le plan image et donc le diagramme de Voronoï de deux segments (b).

Tel que, le diagramme de Voronoï n'est pas complet. En effet, entre les segments, le diagramme de Voronoï n'est pas défini. Si des points de contour se retrouvent dans cette zone, l'affectation à un contour du modèle n'est pas possible. Pour établir le diagramme de Voronoï entre les segments, nous dessinons un cône (à base circulaire et tangent aux plans) à chaque sommet de chaque segment. Nous projetons les cônes dans les images pour déterminer le diagramme (5.19).

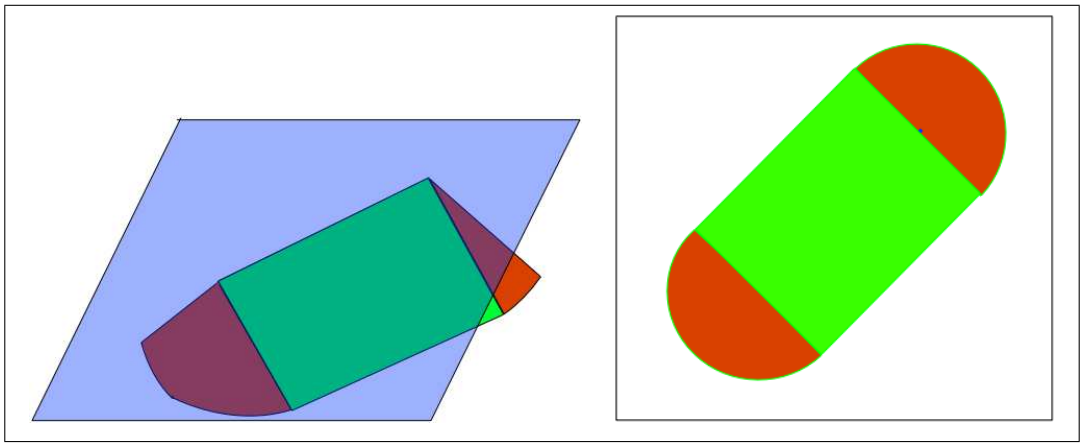


FIG. 5.19: Les régions de Voronoï sont calculées en dessinant des plans de pente 1 et -1 intersectant les segments de droites. Au niveau des sommets des segments, un cône est dessiné pour déterminer la distance entre les segments.

Cette méthode de construction de diagramme de Voronoï est classique. Huttenlocher propose de l'utiliser ([76]) pour effectuer du *template matching*. Elle a été étendue au cas du calcul d'un diagramme de Voronoï en *3D* ([74]).

Calcul de la distance Une fois le diagramme de Voronoï contruit, il s'agit de calculer la distance au contour le plus proche. La méthode standard consiste à utiliser le diagramme de Voronoï pour construire la carte de distances. Il suffit pour cela de lire la profondeur entre le plan contenant les contours et les plans et cônes dessinés en *3D*. La lecture du Z-Buffer de la carte graphique vu du plan de projection du modèle permet d'obtenir automatiquement cette carte des distances. Cependant, comme nous l'avons vu plus haut, la connaissance de la carte ne nous permet pas d'établir une fonction continue et dérivable pour mesurer l'erreur. Nous allons donc établir une distance analytique permettant de calculer sa Jacobienne.

Soit $\boldsymbol{y}$ le vecteur des coordonnées d'un point y du contour extrait des images. Par construction, nous connaissons les coordonnées de ce point dans le repère associé à l'image. Nous voulons calculer la distance de ce point au point du contour modèle le plus proche. La seule difficulté est d'exprimer les coordonnées de y dans un repère où le calcul est aisé. En pratique, nous effectuons un changement de référentiel pour que les coordonnées du point de contour extrait soit exprimées dans le repère associé au contour modèle le proche (c.f. illustration 5.20).

Notons AB le contour modèle et C le point du contour extrait. Les coordonnées de C dans le repère associé à AB sont obtenus en faisant le calcul suivant :

$$\boldsymbol{x}_C^s = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} (\boldsymbol{y} - \boldsymbol{x}_A), \tag{5.15}$$

où $\boldsymbol{x}_C^s$ est le vecteur de coordonnées de C. $\boldsymbol{x}_A$ est le vecteur des coordonnées du point A exprimées dans le repère associé à l'image. Enfin, la matrice est celle d'une rotation d'angle $-\theta$.

Si l'abscisse du point C (dans le repère du segment) est entre 0 et $\|AB\|$ alors l'ordonnée du point correspond à la distance recherchée :

$$d = -(x_{1,C}^i - x_{1,A}^i)\sin(\theta) + (x_{2,C}^i - x_{2,A}^i)\cos(\theta). \tag{5.16}$$

Le cas contraire, la distance est calculée en déterminant la distance entre C et l'extrémité du segment la plus proche :

$$d = \sqrt{(\boldsymbol{y}_1^s - \boldsymbol{x}_{1,A \text{ ou } B})^2 + (\boldsymbol{y}_2^s - \boldsymbol{x}_{2,A \text{ ou } B})^2}. \tag{5.17}$$

Le diagramme de Voronoï du squelette et calcul de l'erreur Nous avons explicité le calcul du diagramme pour un ou deux segments. Nous donnons les résultats obtenus pour le calcul du diagramme de Voronoï du squelette complet. Nous illustrons

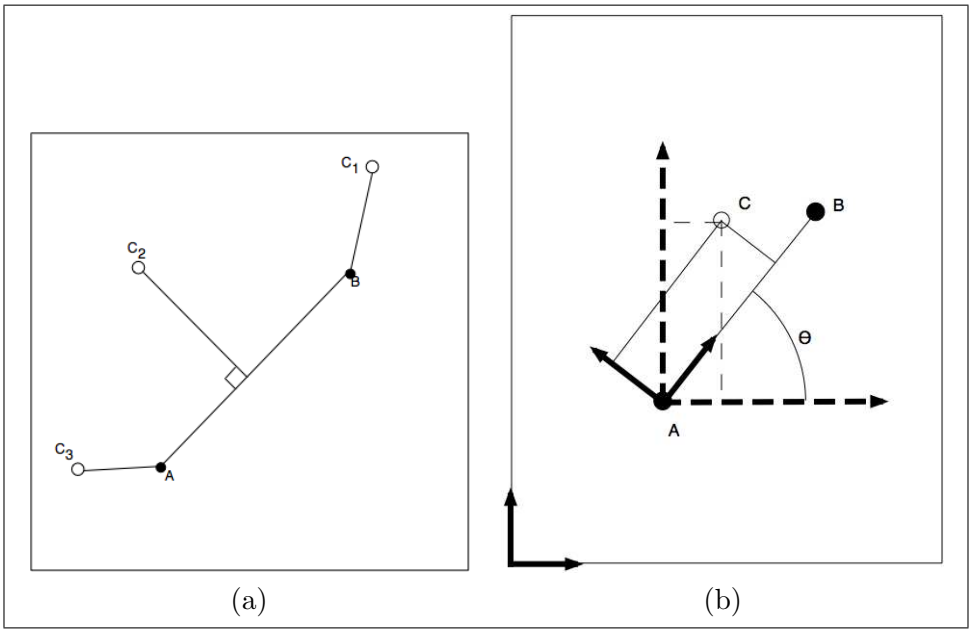


FIG. 5.20: (a) Le calcul de la distance entre un point du contour observé et le contour projeté dépend de la position du point par rapport au segment. Pour calculer aisément la distance, nous effectuons un changement de référentiel. Les variables introduites dans le paragraphe 5.2.2.3 sont illustrées par (b).

dans un premier temps la construction *3D* (c.f. figure 5.21) permettant de calculer le diagramme de Voronoï (c.f. figure 5.22). Enfin, nous donnons à titre indicatif, la transformée en distance obtenue en utilisant le diagramme de Voronoï (c.f. figure 5.23).

Nous avons explicité le calcul de la distance pour un point donné. Nous pouvons maintenant donner la distance orientée du modèle vers l'image :

$$h(\mathcal{Y}, \mathcal{X}) = \sum_{c=0}^{N_c} \sum_{i=0}^{N_e} d^2_{\mathcal{Y}_c}(\boldsymbol{x}^b_{i,c}), \tag{5.18}$$

où N_e est le nombre de points choisis sur le contour image. d est la distance d'un point du contour observé au contour modèle et dont l'expression dépend de la position du point (c.f. plus haut). D'autre part, les points du contour extrait sont choisis de manière aléatoire à chaque itération. Comme [83] le montre, il est plus efficace d'utiliser de l'échantillonnage aléatoire. Cela permet :

- de réduire le nombre de données nécessaires pour effectuer le suivi,
- d'éviter de prendre en compte, pendant toute l'estimation, des points qui peuvent appartenir à des contours parasites,
- d'éviter de choisir des points qui sont en réalité situés trop près d'articulations et donc problématiques (les contours à ces endroits ne sont pas bien définis).

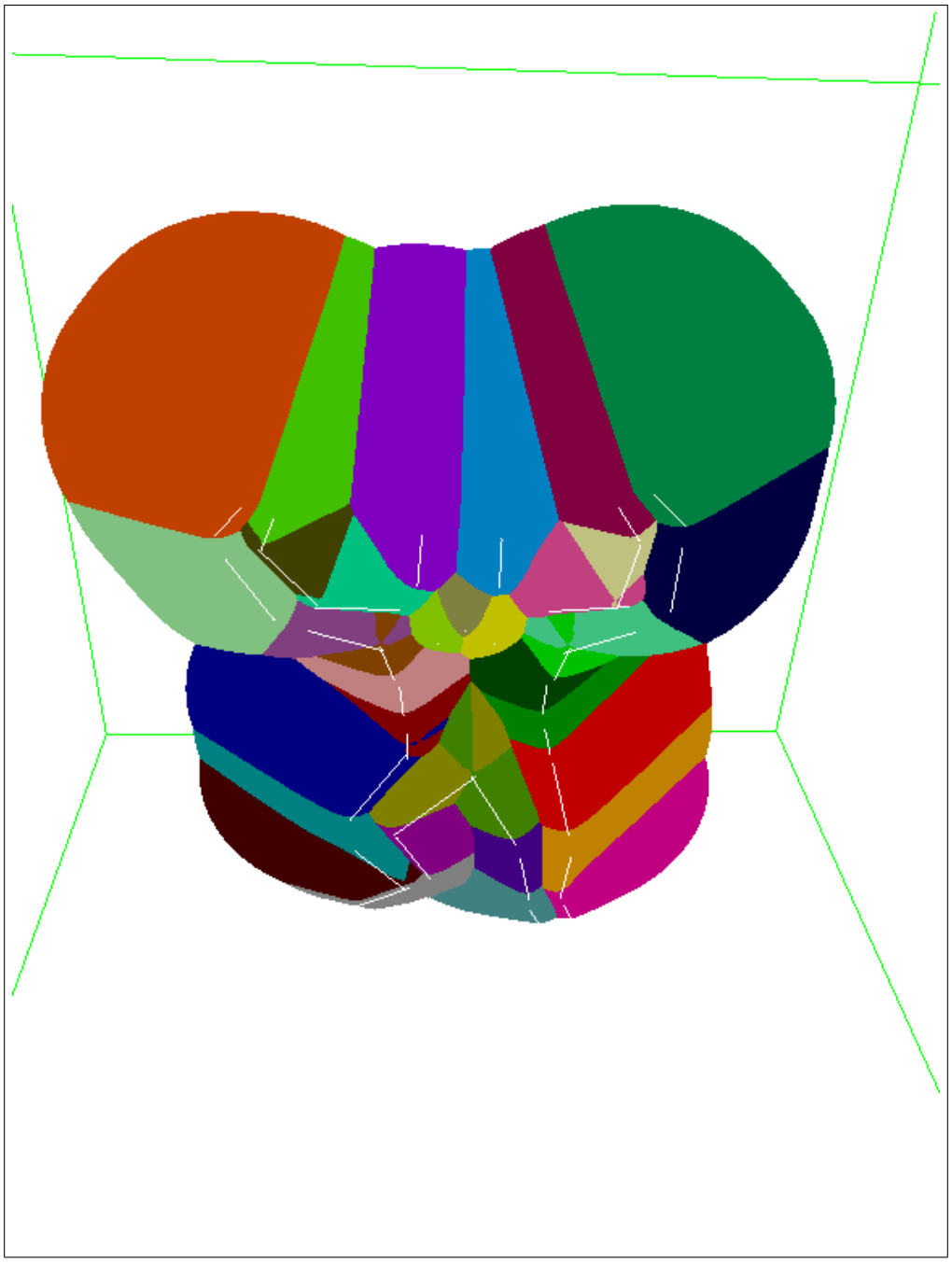


FIG. 5.21: Pour chacun des segments du modèle projeté, 2 cônes et 2 plans sont dessinés. Ce qui nous permet de déterminer la distance de chanfrein ainsi que le diagramme de Voronoï de l'image de contours.

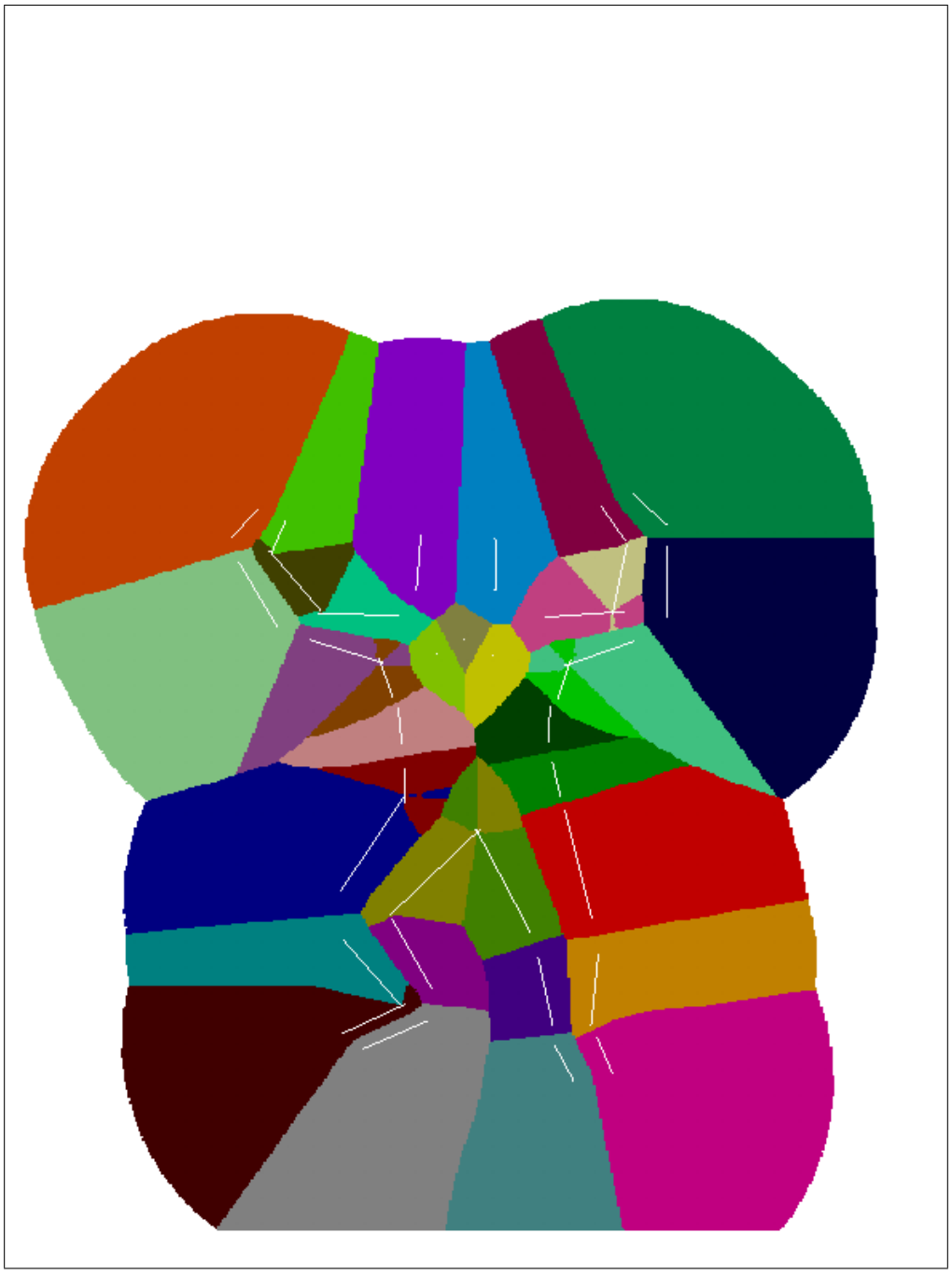


FIG. 5.22: La projection dans le plan du mesh *3D* permet de construire de manière rapide et efficace le diagramme de Voronoï associé au modèle.

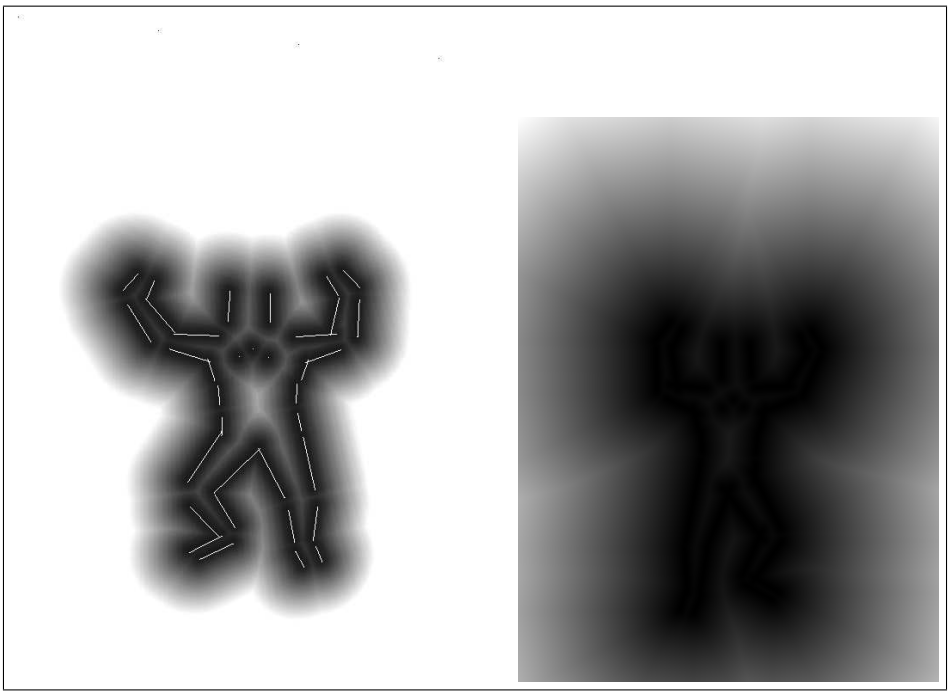


FIG. 5.23: La carte des distances obtenue par le diagramme de Voronoï (à gauche) est comparable à celle utilisant la même méthode qu'avec les contours observés (à droite). La seule différence est le calcul plus localisé pour la distance obtenue à l'aide du diagramme de Voronoï. Les temps de calcul sont donc plus courts pour la carte obtenue à l'aide du diagramme de Voronoï que pour celle obtenue avec la convolution matricielle.

5.2.2.4 Discussions

Nous avons vu que l'utilisation de la transformée en distance permet de calculer rapidement l'erreur du modèle à l'image. Cependant, par construction et par définition, la transformée en distance génère une carte de distances qui est symétrique à proximité des contours. Cette propriété nous pose problème pour l'évaluation de la fonction de coût.

En effet, cette symétrie entraîne des minima locaux dans la fonction de coût à minimiser, lorsque le contour modèle projeté est proche du contour observé. Nous allons montrer pourquoi nous avons ces minima locaux et comment nous pouvons contourner cette difficulté.

Pour la clarté de l'exposé, nous allons nous restreindre au cas idéal d'un cône projeté dans une image. Nous observons donc deux contours qui sont des segments de droite.

Supposons que nous construisions la transformée en distance de la carte des contours du cône (c.f. figure 5.24-(a)). Le long d'une ligne de l'image (en pointillés sur la figure 5.24-(a)), nous traçons le profil de la distance de chanfrein (en trait fin sur la 5.24-(b)). Ce profil est symétrique par rapport aux contours.

Supposons que notre modèle *3D* se projette de sorte à ce que deux des points échantillonnés du contour du modèle *3D* se projettent sur une même ligne de la carte de distance. Nous pouvons tracer le profil de l'erreur à minimiser si nous effectuons une translation du cône le long de la ligne tracée. Nous obtenons le profil de l'erreur représenté sur la figure 5.24-(b) en gras.

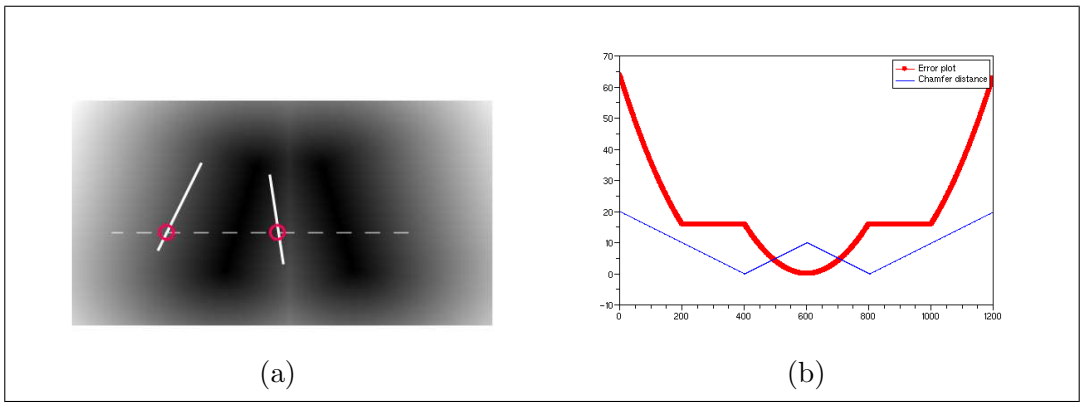


FIG. 5.24: (a) Carte des distances construite à partir de la projection d'un cône dans l'image. (b) Nous traçons le profil de la distance (en trait fin) et de l'erreur à minimiser le long d'une ligne de la carte des distances.

Nous pouvons constater que l'erreur à minimiser contient des minima locaux pour certaines positions des points. Ces minima entraînent un arrêt de la convergence lors de l'estimation des paramètres de pose.

Si nous traçons le même type de profil cette fois-ci en utilisant non plus les contours mais la silhouette, nous obtenons les tracés de la figure 5.25-(b). Nous pouvons voir qu'il

n'y a plus de minima locaux. Nous avons cependant souligné le fait que nous ne pouvions pas utiliser les silhouettes seules pour effectuer la minimisation (c.f. paragraphe 5.1.1).

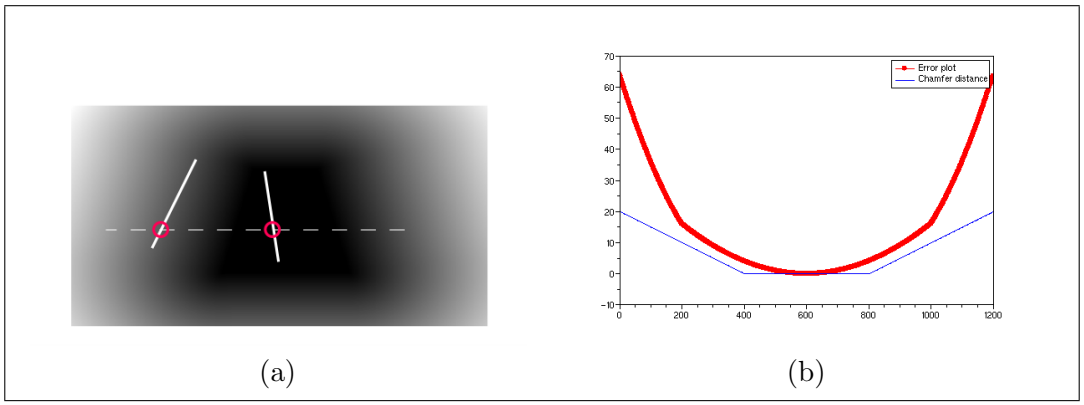


FIG. 5.25: (a) Carte des distances construite à partir de la silhouette d'un cône dans l'image. (b) Nous traçons le profil de la distance (en trait fin) et de l'erreur à minimiser le long d'une ligne de la carte des distances.

Pour éviter les minima locaux qui apparaissent avec l'utilisation des contours, nous utilisons les cartes des distances calculées à partir des silhouettes et des contours. En pratique, nous calculons la carte des distances pour la silhouette et pour les contours séparément et nous additionnons les deux cartes de distances. Nous obtenons alors le profil des distances illustré par la figure 5.26. Il n'y a alors plus de minima locaux.

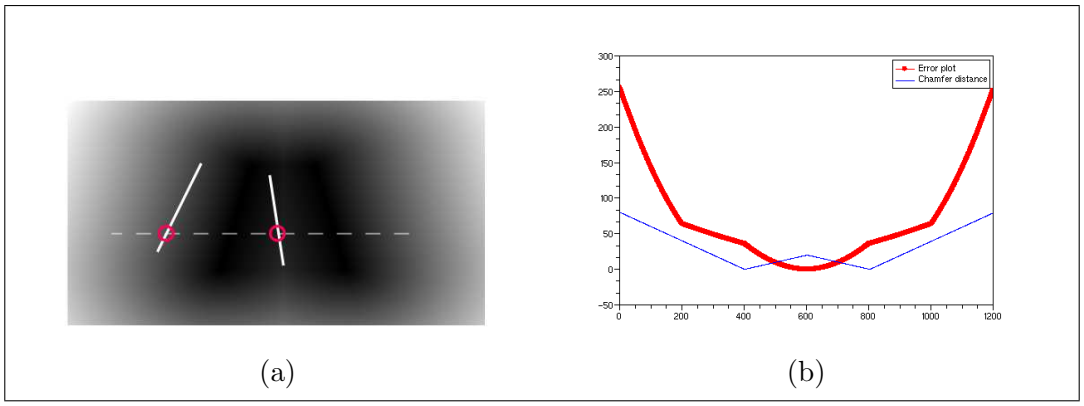


FIG. 5.26: (a) Carte des distances construite à partir de la silhouette d'un cône dans l'image. (b) Nous traçons le profil de la distance (en trait fin) et de l'erreur à minimiser le long d'une ligne de la carte des distances. Contrairement au cas des contours, nous n'avons pas de minima locaux.

Nous avons abordé le cas idéal. Dans le cas réel, nous faisons de même. Nous obtenons donc une nouvelle carte de distances atténuant les effets des minima locaux (figure 5.27).

FIG. 5.27: Somme de la transformée en distance de la carte des contours et des silhouettes.

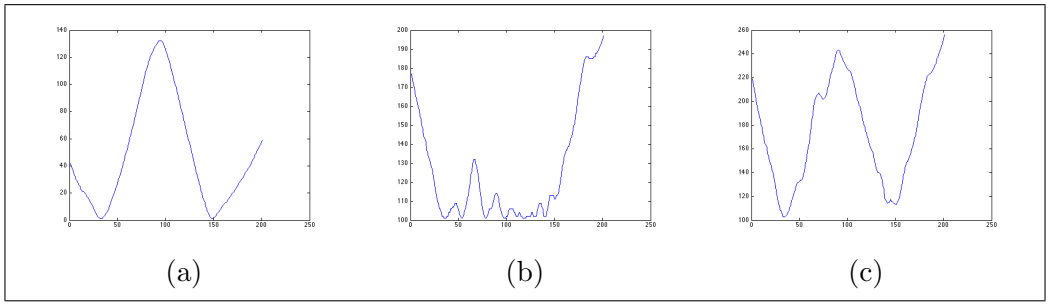


FIG. 5.28: Courbes d'évolution de la distance de chanfrein le long d'une ligne de l'image. (a) est la courbe pour la silhouette. (b) est la courbe pour les contour. Enfin, (c) est la somme des deux courbes. On peut voir sur cette dernière courbe que les minima locaux sont atténués.

Dans le cas de l'extraction des contours basé modèle, le problème de la symétrie est moins gênant. En effet, l'extraction des contours est effectuée pour chaque segment du contour projeté. Contrairement au cas d'un détecteur standard, la carte des distances est calculée pour chaque *patch*. Par construction de la fonction d'erreur, nous avons une symétrie à proximité des contours mais la symétrie n'est pas la même pour les deux contours du cône. L'erreur à minimiser ne présente donc plus de minima.

5.2.3 Synthèse

Afin d'effectuer le suivi de l'acteur dans les images, nous utilisons les silhouettes de l'acteur ainsi que ses contours extraits soit à l'aide d'un détecteur standard de Canny (adapté pour les images couleur) soit à l'aide du détecteur de contours utilisant le modèle. Nous avons montré que l'utilisation des silhouettes seule ne pouvait pas suffire pour effectuer le suivi. Nous avons montré que le détecteur de contour orienté modèle, bien qu'étant une méthode locale, permet d'éliminer les contours parasites (présents avec un détecteur standard). Puis, nous avons explicité le calcul de l'erreur entre les contours extraits de l'image et les contours projetés du modèle. Nous avons vu que nous utilisons la distance de Hausdorff avec une métrique qui dépend, en pratique, de la distance orientée considérée. Pour la première, nous calculons la transformée en distance sur les contours extraits des images et par lecture de la carte, nous déduisons la distance du modèle à l'image. Pour la seconde, nous calculons de manière explicite la distance des points du contour extrait aux points du modèle projeté.

Au final nous obtenons une mesure de l'erreur calculée de manière analytique, qui est continue et dérivable. Elle a pour expression :

$$\boxed{E(\mathcal{X},\mathcal{Y}) = \sum_{c=0}^{N_c}\sum_{b=0}^{N_{bp}}\sum_{i=0}^{2N}\delta_{i,c}^{b} D^2_{\mathcal{Y}_c}(\boldsymbol{x}_{i,c}^{b}) + \sum_{c=0}^{N_c}\sum_{i=0}^{N_e} d^2_{\mathcal{Y}_c}(\boldsymbol{x}_{i,c}^{b}),} \tag{5.19}$$

Intéressons nous maintenant à la minimisation de cette erreur en fonction des paramètres de pose du modèle *3D*.

5.3 Algorithme de suivi

Nous avons vu dans les paragraphes précédents que nous pouvons calculer l'erreur entre l'observation et la prédiction donnée par le modèle de différentes manières. Que ce soit en utilisant la couleur ou les contours, nous avons construit une fonction d'erreur continue et dérivable. La minimisation de ces erreurs permet d'estimer les paramètres du modèle *3D*. Dans cette partie, nous abordons aussi bien l'estimation des paramètres de pose que les paramètres dimensionnels. Nous verrons que ces aspects sont similaires quant à la méthode d'estimation. Il s'agit de résoudre le problème posé dans la partie introductive de ce chapitre :

$$\min_{\boldsymbol{\Upsilon}} E(\mathcal{X},\mathcal{Y}), \tag{5.20}$$

où $\boldsymbol{\Upsilon}$ est l'ensemble des paramètres du modèle *3D*. Nous avons vu qu'en pratique nous séparons le problème du dimensionnement du problème du suivi. Cependant, les fonctions d'erreur dans chacun des cas sont très similaires. Nous pouvons donc adopter le même schéma de minimisation pour ces deux étapes.

Dans un premier, nous introduirons le principe de l'algorithme de Levenberg-Marquardt (LM) que nous utilisons pour minimiser les fonctions de coût. Dans un second nous expliciterons les matrices Jacobiennes des fonctions à minimiser.

5.3.1 L'algorithme de minimisation

Nous avons établi les fonctions de coût selon que nous utilisons la couleur ou les contours dans les images. Ces fonctions dépendant de manière fortement non linéaire des paramètres de pose ou dimensionnels de l'acteur. En outre, ces fonctions sont continues et dérivables. Pour minimiser ces erreurs, nous utilisons donc une méthode de minimisation non linéaire. Parmi celles-ci, nous pouvons citer les méthodes de descente de gradient comme les méthodes de Newton ou de Quasi-Newton, mais aussi des méthodes de type Gauss-Newton (GN), de Levenberg-Marquardt (LM) ou encore des approximations de Quasi-Newton (QN). Pour plus de détails sur ces méthodes de minimisation, le lecteur pourra se référer à [55] et plus précisément aux paragraphes 4.5 et 4.7.

Dans notre cas, nous avons la possibilité de résoudre un problème aux moindres carrés. La méthode de LM ou de QN sont très bien adapté à ce type de problème.

En pratique, nous utilisons la méthode de Levenberg-Marquardt. Comme toutes les autres méthodes de minimisation, cet algorithme est itératif. Une solution initiale $\boldsymbol{\Phi}_0$ est choisie. A chaque itération de la minimisation, nous effectuons la mise à jour des paramètres de la manière suivante : $\boldsymbol{\Phi}_{i+1} = \boldsymbol{\Phi}_i + \boldsymbol{\delta\Phi}$.

Pour déterminer $\boldsymbol{\delta\Phi}$, la fonction de coût $E(\mathcal{X}, \mathcal{Y})$ est linéarisée au premier ordre. Pour simplifier les notations, nous allons poser $S(\boldsymbol{\Phi}) = E(\mathcal{X}, \mathcal{Y}) = \sum f^2(\boldsymbol{\Phi})$. La linéarisation du problème au premier ordre est donc de la forme :

$$f(\boldsymbol{\Phi} + \boldsymbol{\delta\Phi}) = f(\boldsymbol{\Phi}) + \mathbf{J}\boldsymbol{\delta\Phi}, \tag{5.21}$$

où $\mathbf{J}$ est la Jacobienne de la fonction f au point $\boldsymbol{\Phi}$.

Au point solution ($\boldsymbol{\Phi}^*$), c'est-à-dire lorsque S est minimum, nous avons $\nabla S(\boldsymbol{\Phi}^*) = 0$. Si nous différencions le second terme de l'équation (5.21), et que nous nous plaçons à la solution du problème, nous avons :

$$(\mathbf{J}^\top \mathbf{J})\boldsymbol{\delta\Phi} = -\mathbf{J}^\top f. \tag{5.22}$$

Cette équation nous permet de calculer le pas $\boldsymbol{\delta\Phi}$. L'avantage et le point clef de l'algorithme de LM est de modifier le terme de gauche en le régularisant par un facteur λ. L'équation (5.22) devient alors :

$$(\mathbf{J}^\top \mathbf{J} + \lambda \mathbf{I})\boldsymbol{\delta\Phi} = -\mathbf{J}^\top f. \tag{5.23}$$

La coefficient λ est calculé à chaque itération de la minimisation. Le choix est effectué en fonction de la vitesse de convergence. Plus la vitesse est rapide, plus λ sera choisi petit, ceci pour s'approcher de la méthode de type GN. Si la descente ralentit, λ est augmenté pour se rapprocher de la méthode de type descente de gradient.

Le critère d'arrêt de la minimisation est généralement pris de deux manières : soit les paramètres n'évoluent plus au cours des itérations, soit la fonction de coût n'évolue plus.

Cette méthode fait partie des méthodes à région de confiance et nécessite une fonction continue, dérivable avec une Jacobienne bien conditionnée. De plus, la solution initiale doit être « proche » de la solution à estimer. Nous nous plaçons dans le cadre d'une approche de suivi de mouvement. Nous pouvons donc utiliser la pose estimée à l'image précédente de la séquence pour prédire la pose à l'image courante et donc être proche de la pose estimée.

Nous allons maintenant nous intéresser au calcul de la Jacobienne de la fonction de coût que ce soit dans le cadre de l"estimation des paramètres de pose que de l'estimation des dimensions du modèle *3D*.

5.3.2 La Jacobienne des fonctions de coût

Calcul de la Jacobienne associé aux contours Nous pouvons l'estimer soit de manière numérique soit de manière analytique. Etant donné les développements dans le chapitre précédent, nous pouvons calculer explicitement la Jacobienne.

Pour cela, nous allons dériver les équations (5.9) et (5.19).

Pour calculer la Jacobienne de (5.19), nous calculons $\frac{dE}{dt} = \frac{dE}{d\boldsymbol{x}}\frac{d\boldsymbol{x}}{dt}$, où $\boldsymbol{x}$ est un point du contour extrémal. Nous pouvons donc écrire :

$$\begin{aligned}\frac{dE}{dt} &= \frac{dE}{d\boldsymbol{x}}\frac{d\boldsymbol{x}}{dt} \\ &= \frac{dE}{d\boldsymbol{x}}\mathbf{J}_I(\mathbf{A}+\mathbf{B})\mathbf{J}_H\dot{\mathbf{\Phi}}. \end{aligned} \tag{5.24}$$

Il reste à calculer :

$$\frac{dE}{d\boldsymbol{x}} = \frac{dh(\mathcal{X},\mathcal{Y})}{d\boldsymbol{x}} + \frac{dh(\mathcal{Y},\mathcal{X})}{d\boldsymbol{x}}. \tag{5.25}$$

Nous pouvons réécrire chacun des termes de la manière suivante :

$$\frac{dh(\mathcal{X},\mathcal{Y})}{d\boldsymbol{x}} = \sum_{c=0}^{N_c}\sum_{b=0}^{N_{bp}}\sum_{i=0}^{2N}\delta_{i,c}^b\frac{dD^2_{\mathcal{Y}_c}(\boldsymbol{x}_{i,c}^b)}{d\boldsymbol{x}}, \tag{5.26}$$

$$\frac{dh(\mathcal{Y},\mathcal{X})}{d\boldsymbol{x}} = \sum_{c=0}^{N_c}\sum_{i=0}^{N_e}\frac{dd^2_{\mathcal{Y}_c}(\boldsymbol{x}_{i,c}^b)}{d\boldsymbol{x}}. \tag{5.27}$$

Il reste donc à déterminer $\frac{dD_{\mathcal{Y}_c}}{d\boldsymbol{x}}$ et $\frac{dd_{\mathcal{Y}_c}}{d\boldsymbol{x}}$.

– $\frac{dD_{\mathcal{Y}_c}}{d\boldsymbol{x}}$. Il s'agit d'une interpolation bilinéaire à différencier. Nous avons :

$$\begin{aligned}\frac{\partial D_{\mathcal{Y}_c}(x_1)}{\partial x_1} &= s(C_{\mathcal{Y}}(u+1,v+1) - C_{\mathcal{Y}}(u,v+1)) \\ &\quad +(1-s)(C_{\mathcal{Y}}(u+1,v) - C_{\mathcal{Y}}(u,v)) \qquad (5.28)\\ \frac{\partial D_{\mathcal{Y}_c}(x_2)}{\partial x_2} &= r(C_{\mathcal{Y}}(u+1,v+1) + C_{\mathcal{Y}}(u,v+1)) \\ &\quad +(r-1)(C_{\mathcal{Y}}(u+1,v) + C_{\mathcal{Y}}(u,v)). \qquad (5.29)\end{aligned}$$

Pour calculer ces deux dérivées, nous faisons l'hypothèse que $d[x]/dx = 0$. En fait la fonction partie entière est dérivable par morceaux et non dérivable si $[x] = x$. Cette égalité apparaît si les contours *3D* du cône se projettent sur un pixel (de manière exacte). C'est un cas que nous pouvons négliger lors de notre dérivation.

– $\frac{dd_{\mathcal{Y}_c}}{d\boldsymbol{x}}$. Pour calculer ce terme, il suffit de dériver les équations (5.16) et (5.17), ce qui est aisé.

Calcul de la Jacobienne associé à la couleur La Jacobienne de la fonction de coût (5.9) se calcule de manière similaire à celle que nous avons explicitée ci-dessus. Il s'agit de dériver une interpolation bilinéaire. Nous pouvons donc appliquer les relations établies dans le cadre de l'interpolation pour la lecture de la distance de chanfrein (c.f. equations (5.28) et (5.29)).

Remarque pratique : L'algorithme de Levenberg-Marquardt que nous utilisons prend comme entrées le vecteur $\boldsymbol{v}_E = (\{D_{\mathcal{Y}_c}(\boldsymbol{x}^b_{i,c})\}_{\{i,c,b\}}, \{d_{\mathcal{Y}_c}(\boldsymbol{x}^b_{i,c})\}_{\{i,c,b\}})$ de dimension $m = N_c \times N_{bp} \times N$, ainsi que la Jacobienne de ce vecteur, $\mathbf{J}_{\boldsymbol{v}_E}$ qui est de dimensions $m \times p$, où p est le nombre de paramètres à estimer.

5.3.3 Discussion

Dans cette section, nous avons abordé l'estimation des paramètres du modèle *3D*. Nous avons vu que nous utilisions l'algorithme de Levenberg-Marquardt pour effectuer la minimisation. En pratique, cette algorithme est très efficace. Cependant, il ne permet pas de prendre en compte des contraintes dans la minimisation. L'estimation des paramètres de pose, notamment, pourrait être contraints pour n'avoir qu'un débattement limité. Pour prendre en compte ces contraintes, nous pouvons utiliser d'autre méthodes comme celle de BFGS ou encore des méthode de type points intérieurs qui permettent de résoudre un problème de minimisation sous contrainte. Nous avons décidé de ne pas utiliser de telle méthode car nous pensons que pour notre cas particulier, les articulations ne doivent pas être contraintes. Nous estimons un mouvement effectué par un acteur réel, le résultat de l'estimation doit donc aussi être correct. Le cas contraire, c'est que la minimisation est erronée. La conséquence est qu'au cours de la minimisation, nous laissons la possibilité à l'algorithme de passer par des configurations interdites (bio-mécaniquement impossibles).

Enfin, nous avons choisi de calculer la Jacobienne de manière analytique, car nous avons constaté que le calcul de la Jacobienne par méthode numérique ne donnait pas toujours les résultats escomptés lors de la minimisation. En effet, la fonction de coût est fortement non linéaire par rapport aux différents paramètres du modèle.

5.4 Initialisation du suivi

Dans la partie précédente, nous avons abordé la problématique du suivi du mouvement. Pour pouvoir effectuer ce suivi, nous avons mis en place un méthode locale de recherche des paramètres. Cela est possible, car nous pouvons utiliser la pose estimée à l'image précédente pour initialiser le suivi à l'instant courant. Cela suppose aussi que pour la première image de la séquence vidéo, la pose est correcte. Nous allons maintenant aborder le problème de l'estimation de la pose dans la première image de la séquence.

5.4.1 Problématique

Dans le protocole opératoire pour effectuer une séance de capture, nous demandons à l'acteur de démarrer la séquence du mouvement à capturer dans une pose donnée. Cette pose initiale nous permet de simplifier l'amorce du suivi du mouvement. Cependant, la pose n'est pas strictement contrainte : l'acteur peut se situer n'importe où dans la scène et la pose initiale peut légèrement varier d'une séquence à l'autre. Nous avons donc dû mettre en place un protocole particulier pour l'initialisation de la pose.

Le problème principal est que la pose initiale du modèle se trouve assez loin de la pose réelle de l'acteur. Nous devons donc adapter la méthode d'estimation de la pose à ce problème. En effet, effectuer l'estimation de la pose en laissant libre l'ensemble des paramètres articulaires de la chaîne cinématique peut amener à un échec de l'estimation (c.f. figure 5.4.1-(b)).

Il s'agit alors d'adopter une approche hiérarchique pour initialiser la pose de l'acteur.

5.4.2 Une approche hiérarchique

Nous allons décrire maintenant la méthode d'initialisation de la capture du mouvement. Nous organisons la chaîne cinématique en niveaux hiérarchiques (c.f. illustration 5.30). Pour effectuer l'estimation de la pose initiale, nous débloquons de manière successive l'ensemble des niveaux hiérarchiques. Nous estimons donc dans un premier temps le déplacement rigide (de l'ensemble du modèle) en ne laissant libre que les d.d.l. du pelvis. Puis nous libérons au fur et à mesure les d.d.l. dans la hiérarchie des articulations.

Nous pouvons soit utiliser la couleur soit les contours. Cependant, l'utilisation de la couleur n'est pas recommandée. En effet, il s'agit d'une méthode locale, puisque les

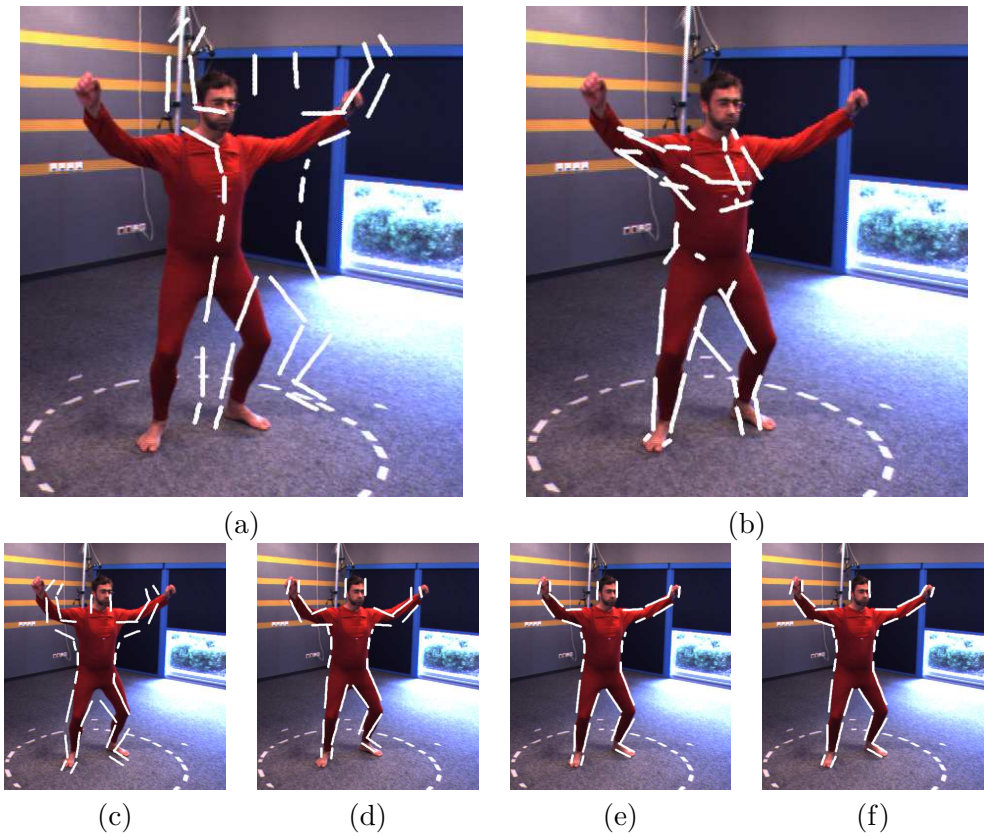


FIG. 5.29: Nous illustrons l'initialisation de la pose avant d'effectuer la capture du mouvement. Si nous effectuons l'estimation sans prendre de précautions nous aboutissons à un échec (b). Cependant, une approche hiérarchique permet d'initialiser correctement la pose de l'acteur au début du suivi du mouvement. Seuls les 6 d.d.l. du modèle sont libres lors de la première estimation (c). Puis les d.d.l. sont relâchés de manière hiérarchique (d,e,f). L'estimation de la pose est alors correcte.

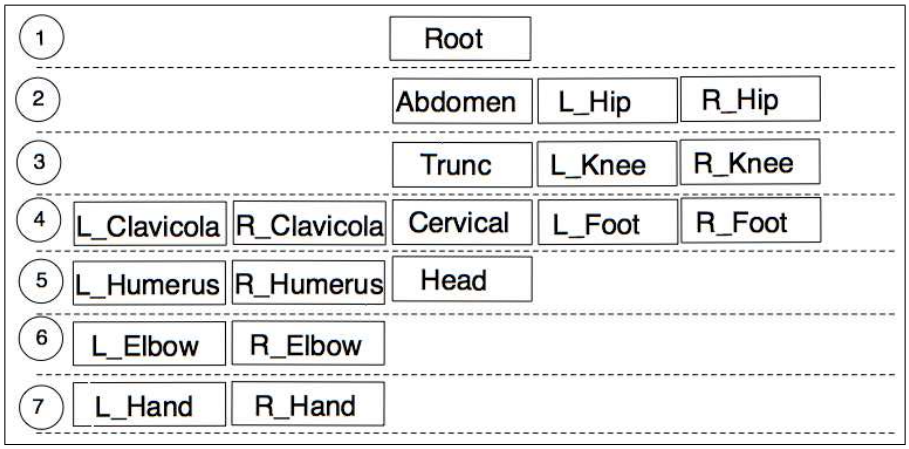


FIG. 5.30: Le squelette est organisé de manière hiérarchique. Cette organisation permet lors de l'initialisation du mouvement, d'estimer correctement la pose.

contraste de couleur doivent être choisis pour correspondre aux contours extrémaux de l'acteur. La recherche des bons contrastes de couleur peut s'avérer coûteuse si le modèle n'est pas proche de la pose de l'acteur. Pour aider à l'initialisation, l'acteur démarre la séquence avec une pose de type « akah ». Celle-ci permet de rendre moins ambiguë l'estimation de la pose initiale d'une part et l'estimation des dimensions des cônes comme nous le verrons dans la section suivante.

5.5 Dimensionnement du modèle

Pour effectuer la capture du mouvement, il est nécessaire, tout du moins dans le cadre de nos travaux, que le modèle *3D* soit correctement dimensionné pour que celui-ci corresponde au mieux à l'acteur. Il est donc important dans une première étape d'effectuer un ajustement des différents paramètres de notre modèle *3D*. Il s'agit de déterminer l'ensemble des dimensions de chacune des primitives pour que la projection de celles-ci dans les images se superposent correctement sur l'acteur filmé. Nous devons donc déterminer d'une part la longueur des segments de la chaîne cinématique mais aussi les dimensions de chacun des cônes.

Il y a donc deux étapes nécessaires pour calibrer le modèle *3D*. Dans un premier temps, il faut ajuster le squelette du modèle, pour que la pose de celui-ci corresponde à la pose de l'acteur. Dans un second temps, les dimensions des cônes sont ajustées pour que la représentation *3D* corresponde au mieux à l'acteur. Cependant, ces deux étapes ne sont pas tout à fait décorrélées. En effet, l'ajustement des dimensions nécessite d'affiner les paramètres de pose, et la pose peut d'autant mieux être estimée, que le modèle *3D* est correct. Nous procédons donc en alternant l'estimation des paramètres de pose avec l'estimation des dimensions. L'estimation de tous les paramètres simultanément est réalisable mais peut poser quelques problèmes dont nous discuterons en fin de partie.

Dans un premier temps nous justifierons le fait de dimensionner le modèle en utilisant une méthode semi-automatique (c.f. paragraphe 5.5.1). Puis nous détaillerons l'initialisation du squelette (c.f. 5.5.2) ainsi que le dimensionnement du modèle *3D* (c.f. 5.5.3). Enfin, nous discuterons d'une méthode robuste d'estimation des paramètres.

5.5.1 Approche

Quasiment toutes les méthodes actuelles de suivi de mouvement utilisent un modèle *3D* dont les dimensions ont été ajustées à la main par un utilisateur. Une seconde solution, aussi souvent utilisée, est d'utiliser des appareils de mesure externe (comme un scanner) pour évaluer les dimensions de l'acteur, puis d'intégrer ces mesures dans le système de capture par vision. Enfin, quelques auteurs proposent de dimensionner ou de construire le modèle automatiquement à partir des observations. Nous pouvons citer par exemple [82] ou encore [140], où le modèle est créé à partir des observations.

Cependant, l'ajustement manuel ou à l'aide d'appareils externes ne nous semble pas optimal pour le suivi de mouvement. L'ajustement des paramètres à l'aide de méthodes spécifiques ne nous semble pas non plus approprié. En effet, les dimensions du modèle devraient être adaptés aux besoins de l'algorithme. L'adaptation manuelle de l'ensemble des dimensions n'est en fait pas optimale. Une explication peut être que l'homme adapte au mieux à ce qui est observé, alors que les algorithmes de traitement génèrent un bruit d'observation qui perturbe l'estimation. Le dimensionnement automatique prend alors en compte cette perturbation. Nous avons donc mis en place une méthode semi-automatique de dimensionnement de l'ensemble du modèle *3D* à l'aide de techniques équivalentes à celles qui nous utilisons lors du suivi du mouvement humain.

De même que pour le suivi de mouvement, il s'agit de mettre en correspondance les contours extraits des images avec les contours du modèle projetés dans les images. Les contours images doivent être sélectionnés avec attention. En effet, il s'agit d'associer les contours modèle avec les contours extrémaux de l'acteur et non des contours liés par exemple à des plis de vêtement. De la même manière que pour le suivi, les contours recherchés sont ceux qui maximisent la différence entre la couleur de l'acteur et la couleur du reste de l'image.

5.5.2 Estimation du squelette

Un squelette générique est utilisé pour initialiser l'algorithme. Il s'agit uniquement d'une description cinématique du modèle. L'utilisateur positionne le squelette manuellement pour que les articulations du modèle coïncident avec celles de l'acteur. Cette opération s'effectue en cliquant sur les articulations dans différentes images. Par triangulation et optimisation, les positions des différentes articulations sont reconstruites en *3D*. Une fois les positions reconstruites, le squelette générique est adapté pour correspondre aux points reconstruits. La longueur des segments de la chaîne articulaire et les paramètres articulaires sont adaptés. Cette étape est effectuée en minimisant la distance euclidienne entre les point *3D* reconstruits et les articulations du modèle. Cette dernière

étape est nécessaire pour estimer la pose de l'acteur. En effet, la simple connaissance de la position *3D* des articulations ne permet pas de déterminer la pose de l'acteur. Plus précisément, le squelette a des contraintes articulaires comme par exemple des d.d.l. limités pour certaines articulations. Il faut estimer l'ensemble des paramètres articulaires pour que la pose du squelette, étant donné les contraintes articulaires, corresponde au mieux aux points sélectionnés par l'utilisateur. Cette estimation est effectuée par optimisation des paramètres de pose pour minimiser la distance entre les centres articulaires reconstruits et les centres articulaires du modèle générique.

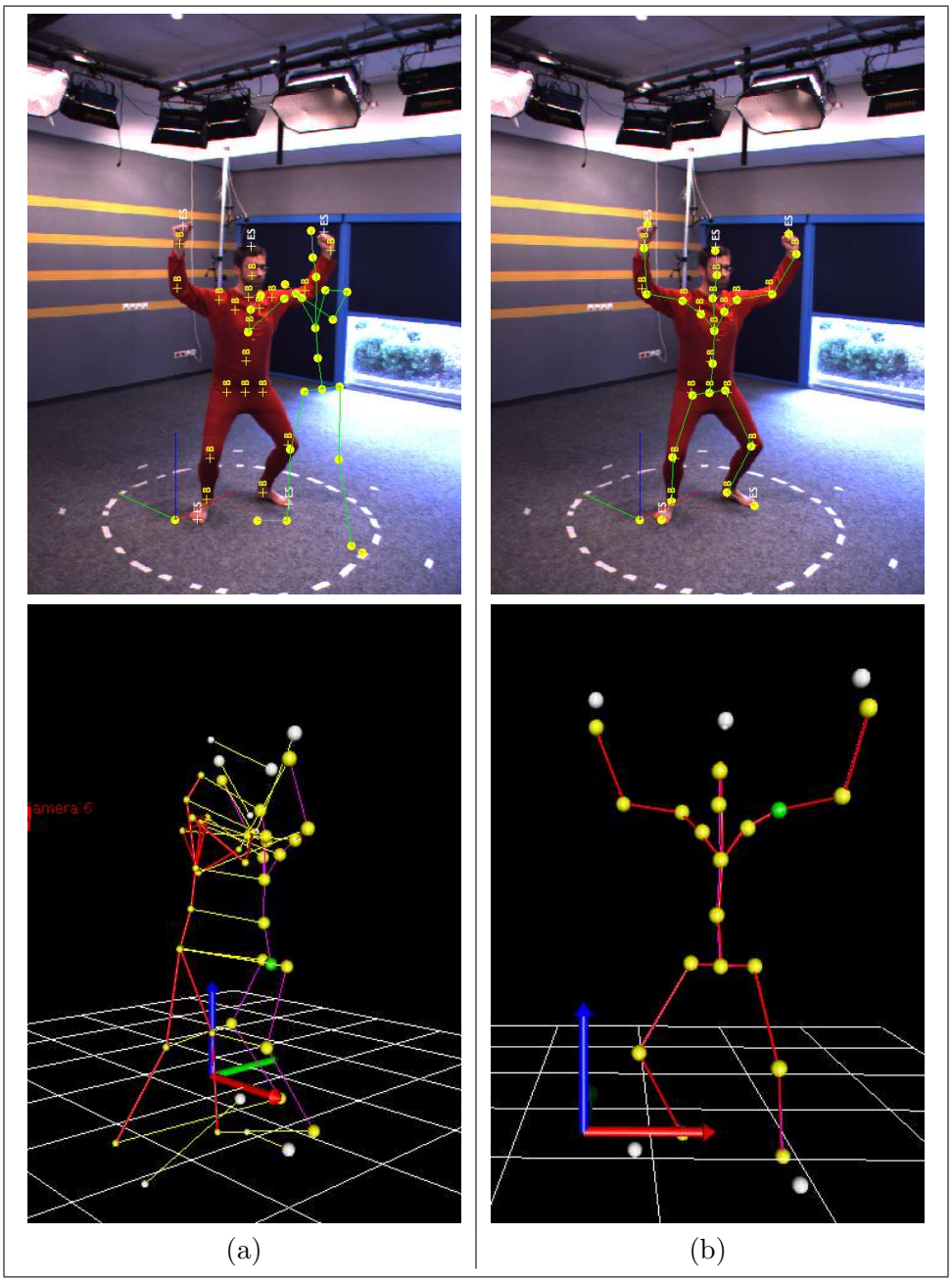
5.5.3 Dimensionnement du modèle *3D*

Cette seconde étape est la plus complexe des deux. Il s'agit de dimensionner l'ensemble des cônes du modèle *3D*, pour que ce dernier corresponde au mieux à l'acteur. Comme nous l'avons vu précédemment, nous utilisons la couleur de la même manière que dans le cadre du suivi. Cependant, au lieu de minimiser les paramètres de pose de l'acteur, nous minimisons les paramètres dimensionnels.

Cependant, l'utilisation des variances sur la couleur pose des difficultés que nous exposons maintenant.

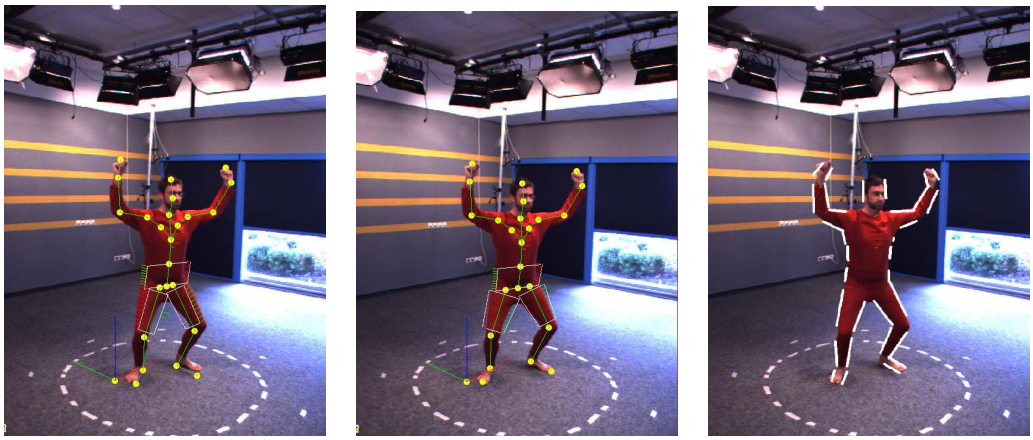
Les vêtements : Lors de l'estimation des dimensions de l'acteur, il est important de faire attention aux vêtements. En effet, nous cherchons à modéliser l'acteur au mieux pour que lors du mouvement, les contours des cônes dans les images correspondent au mieux à l'observation de l'acteur. Les vêtements changent de forme au cours du mouvement. Le deuxième problème est la couleur du vêtement, qui, si elle est homogène ou se distingue mal du fond de l'image, empêche une estimation correcte des dimensions. Pour contourner la première difficulté liée aux plis du vêtement, nous effectuons la minimisation sur plusieurs images en même temps. Pour le deuxième problème, nous essayons de choisir des vêtement qui se contrastent au mieux avec le fond de l'image. Le problème des couleurs homogènes n'apparaît pas lors de la phase de capture de l'acteur. En effet, la pose générique requise (« haka ») pour cette étape évite les superpositions des différentes parties du corps. Le problème se pose lors de l'utilisation de la méthode pour le suivi du mouvement.

Les occultations et la proximité : L'estimation des dimensions est sensible aux contours parasites ou aux mauvais assignements. Tout comme pour le suivi du mouvement, nous utilisons la détection de la visibilité des contours modèles pour éviter l'affectation des contours modèles à des parties du corps en réalité invisibles. D'autre part, les parties du corps dont nous estimons les dimensions doivent être clairement visibles. Par conséquence, nous éliminons les contours des parties du corps dont l'axe principal est quasiment parallèle à l'axe optique de la caméra considérée. Enfin, lorsque deux contours sont trop proches, ils ne sont pas pris en compte dans le calcul des dimensions. C'est pour toutes ces raisons que, d'une part, la pose de l'acteur est déterminée et que d'autre part, nous faisons la minimisation sur plusieurs images.



(a) (b)

FIG. 5.31: La pose « haka » permet à l'utilisateur de cliquer avec précision sur les articulations pour initialiser la pose du squelette. (a) Le squelette générique est chargé dans l'application et l'utilisateur clique sur les articulations dans les images (croix jaunes). Une reconstruction *3D* est faite. Nous voyons en jaune la correspondance entre les articulations du squelette générique (en rouge) et les articulations du squelette reconstruit (en violet). (b) Après estimation de l'ensemble des paramètres du squelette reconstruit, nous obtenons la pose de l'acteur.



(a) (b) (c)

FIG. 5.32: Les primitives surfaciques sont dimensionnées de manière automatique. Elles sont initialisées avec des valeurs génériques (a). Puis dimensionnées avec la méthode utilisant la couleur (b). Lorsque le modèle est dimensionné, ses contours projetés se superposent correctement à ceux de l'acteur (c).

5.5.4 *Bundle adjustement* sur les paramètres de pose et les dimensions des cônes

Nous avons vu dans les paragraphes précédents que nous estimons les paramètres de pose et les dimensions de l'acteur sur une seule image et de manière séparée.

Nous pouvons en réalité estimer la pose ainsi que les dimensions du modèle de manière simultanée. Cependant, la pose du modèle doit être suffisamment proche de la pose de l'acteur. En effet, l'estimation des paramètres dimensionnels nécessite de faire l'hypothèse que les contours recherchés ne soient pas trop loin des contours projetés. La pose initiale est donc estimée dans un premier temps avec la technique décrite ci-dessus (méthode semi-automatique avec intervention de l'utilisateur). Puis, lors de l'estimation des dimensions des différentes parties du corps, les paramètres articulaires sont laissés libres. Cette liberté permet à l'algorithme de recaler les articulations si besoin est, pour affiner l'estimation des dimensions.

Pour améliorer l'estimation des dimensions de l'acteur, nous pouvons aussi utiliser plusieurs images. En effet, les contours observés dans les images dépendent de la pose de l'acteur, des vêtements portés et des traitements effectués pour extraire les contours images. Le fait d'estimer les paramètres sur plusieurs images permet de réduire l'effet de ces perturbations. La pose est donc ré-estimée pour chaque nouvelle image en fixant les paramètres de dimensions. Puis ces paramètres sont relâchés pour affiner le dimensionnement.

Deuxième partie

Expérimentations et Extensions

Chapitre 6

Résultats

Sommaire

Résumé

Dans un premier temps, nous décrivons le système matériel que nous avons utilisé pour effectuer la capture du mouvement. Puis, nous décrivons l'ensemble des applications mises en oeuvre pour effectuer la capture du mouvement. Ensuite, nous décrivons les expérimentations et nous donnons les résultats du projet SEMOCAP obtenus. Nous montrons les résultats obtenus par l'UHB et comparons ces derniers à ceux obtenus avec un système à marqueurs de type VICON. Enfin, nous discutons des conditions de réussite de l'estimation des paramètres de pose.

Introduction au chapitre

Dans ce chapitre, nous allons décrire la mise en oeuvre pratique des développements théoriques proposés dans les précédents chapitres. Dans le cadre du projet SEMOCAP, nous avons mis en place une plate-forme matérielle permettant de filmer les acteurs avec plusieurs caméras synchronisées ainsi qu'une plate-forme logicielle permettant aux protagonistes du projet d'utiliser les divers algorithmes développés pour calibrer les caméras, effectuer la soustraction de fond, dimensionner le modèle *3D*, estimer les trajectoires angulaires de l'acteur et analyser les trajectoires pour les appliquer à un personnage virtuel. Nous développons ces points dans la première partie. Dans un second temps, nous proposons des résultats obtenus sur des séquences synthétiques et des séquences réelles avec les différentes méthodes décrites au chapitre 5. Enfin, nous discutons des conditions nécessaires pour effectuer le suivi du mouvement ainsi que de critères d'évaluations des résultats obtenus.

6.1 Mise en oeuvre expérimentale

Dans cette partie, nous décrivons l'ensemble du dispositif expérimental que nous avons mis en place pour effectuer la capture du mouvement. Dans le cadre du projet SEMOCAP, l'INRIA a développé les logiciels d'une chaîne complète de capture des acteurs et de leurs mouvements, sur une architecture matérielle développée par ASICA. Les développements nécessaires sont une part importante du travail réalisé pendant la thèse. En effet, la société ARTEFACTO doit pouvoir utiliser l'ensemble des développements pour utiliser le système de capture du mouvement. Nous avons donc fait en sorte de fournir des développements utilisables par les protagonistes du projet.

6.1.1 Le matériel

Le système est constitué de plusieurs caméras IEEE1394 industrielles (nombre variable) connectées chacune à un MiniPC. Chaque MiniPC est connecté par réseau Ethernet à un poste principal (poste maître). Le poste maître contient une IHM (Interface Homme-Machine) permettant de régler chaque caméra, de visualiser les flux vidéos et de les enregistrer, à la manière d'une « régie vidéo ».

Sur la figure 6.1, nous remarquons que chaque caméra est reliée à un boîtier de synchronisation. Les caméras utilisées offrent une possibilité de synchronisation par une source externe. Le boîtier de synchronisation envoie à intervalles réguliers un signal qui déclenche la capture d'une image sur chaque caméra. Le flux vidéo est enregistré sur le MiniPC relié à la caméra.

Les caméras sont disposées sur des supports mobiles permettant une grande flexibilité pour la configuration de l'espace d'acquisition. Afin de minimiser les pertes de

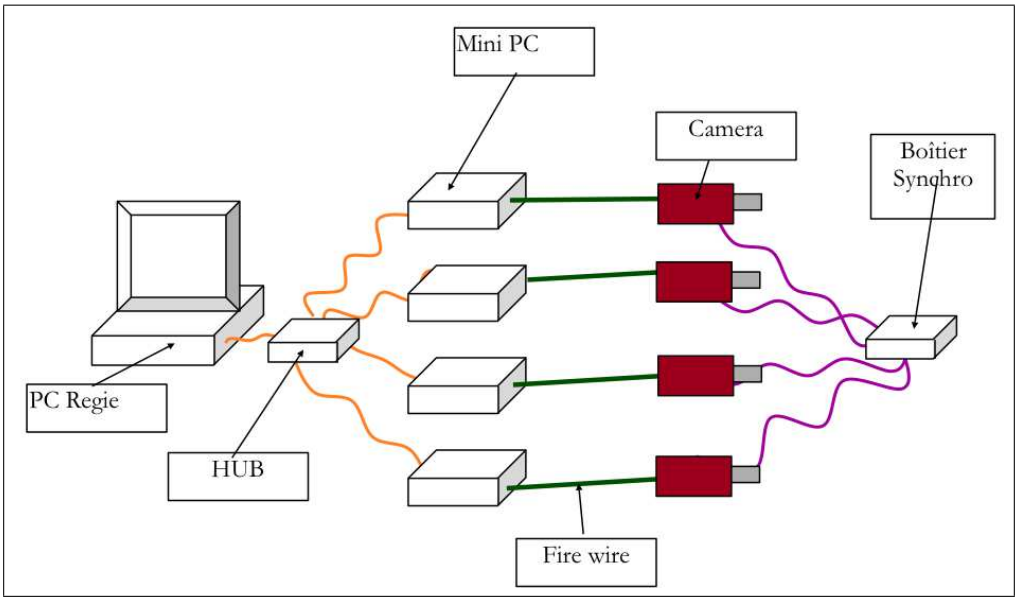


FIG. 6.1: Dans le cadre du projet SEMOCAP, nous avons utilisé quatre caméras FireWire accompagnées d'un MiniPC. Chacune des caméras est synchronisée par un boîtier externe et commandée par un PC maître permettant aussi de visualiser les données acquises.

signal, les PC (ou MiniPC) sont disposés à moins de cinq mètres des caméras. La flexibilité est alors au niveau du câblage entre les différents PC et le poste maître. Le PC maître est donc placé en dehors du champ de vue de l'ensemble des caméras.

L'éclairage de la scène de capture est un point très important. Afin d'avoir un éclairage constant et diffus, nous avons opté pour des Kinoflo. Ces lampes très utilisées lors des tournages pour la télévision permettent de limiter les ombres dans la scène de capture.

Le coût matériel du système dépend du nombre de caméras. Il faut compter environ 2000 € par caméra et PC (ou MiniPC) associé, et environ 2000 € également pour la station maître et le module de synchronisation. Pour le système muni de quatre caméras, le coût matériel du système est de l'ordre de 10 000 €.

Lors de l'installation du matériel, nous devons faire attention aux deux points suivant :

Le volume d'acquisition : A nombre de caméra donné, il existe un compromis entre le volume d'acquisition et la précision avec laquelle la capture du mouvement est faite. En effet, plus le volume d'acquisition est grand, plus le champ de vue des caméras doit être grand et donc plus petite sera la résolution de l'acteur. L'estimation du mouvement s'en retrouvera donc moins précise. Le placement des caméras joue aussi un rôle important. En effet, les différentes caméras doivent être placées de sorte à couvrir au mieux l'acteur. Lors de notre séance d'acquisition à Rennes, nous avions quatre caméras que nous avons placées en demi-cercle de

sorte à maximiser le volume d'acquisition. Cependant, ce placement de caméras a contraint l'acteur à effectuer des mouvements face aux caméras. Dans le cadre de la plate-forme GRIMAGE, nous disposons de six à huit caméras ce qui permet de couvrir un espace beaucoup plus complet. Sur cette dernière plate-forme, nous avons opté pour des acquisitions avec une résolution élevée et donc un volume restreint. Ainsi, des mouvements plus complexes et plus rapides peuvent être capturés.

L'éclairage et l'environnement d'acquisition : pour pouvoir effectuer une soustraction de fond optimale, il est nécessaire d'avoir un éclairage constant de la scène et qui minimise les ombres portées de l'acteur (que ce soit sur le sol ou sur les murs). Un éclairage trop important peut aussi nuire à la visibilité des contours ou des couleurs dans les images. De plus, le fond de l'image doit être statique. Dans le cadre de nos acquisitions à Rennes, cette dernière condition n'était pas remplie, non pas à cause de mouvements dans la pièce mais à cause de l'éclairage de la scène par les spots de LED du système VICON. Le clignotement des LED a rendu la soustraction de fond très bruitée.

6.1.2 Principe de fonctionnement

Depuis le poste maître, l'utilisateur règle les paramètres des différentes caméras (luminosité, saturation, *etc.*) afin de réaliser la capture dans les meilleurs conditions. Toujours à partir du poste maître, l'utilisateur lance ensuite l'enregistrement des séquences. Le format utilisé pour les vidéos est le BLK qui est un format lié à la bibliothèque Blinky (développée en interne à l'INRIA) qui est utilisée dans les développements de l'INRIA.

L'acquisition des séquences vidéo se déroule en plusieurs étapes :

Calibrage du système : Cette phase permet de s'assurer que le système est opérationnel (flux synchronisés et enregistrement des séquences vidéo opérationnel) et d'estimer les paramètres des caméras.

Acquisition du fond : Nous procédons à l'acquisition de la scène vide. Cette séquence sera ensuite utilisée pour l'apprentissage du fond et permettra de construire un modèle de celui-ci. Ce modèle sera par la suite utilisé pour la segmentation des images, c'est-à-dire la séparation de l'acteur du reste de la scène.

Acquisition des séquences : Les mouvements sont acquis et mémorisés sous forme de vidéos.

Traitement des séquences et extraction du mouvement : Plusieurs calculs sont nécessaires à l'extraction du mouvement et à la génération d'un fichier exploitable dans une application *3D*.

Calibrage du système Outre la vérification du bon fonctionnement de l'ensemble du système, cette étape permet de déterminer la position et l'orientation des caméras dans un repère commun ainsi que les paramètres internes de chacune d'elles (la focale et la distorsion). Afin de déterminer tous ces paramètres, nous utilisons un bâton doté de quatre marqueurs lumineux (pour être facilement vu lors des traitements),

dont les positions relatives sont connues précisément. Trois séquences sont enregistrées. Les deux premières, au cours desquelles le bâton est posé au sol, permettent de fixer le référentiel du monde dans lequel l'ensemble des coordonnées des caméras seront données. La troisième séquence est un mouvement du bâton dans le volume d'acquisition. Elle permet de déterminer l'ensemble des paramètres (extrinsèques et intrinsèques) des caméras. Nous présentons quelques captures d'écran dans l'annexe **??**.

Acquisition des séquences de fond Il est nécessaire de connaître l'environnement de capture pour extraire la silhouette de l'acteur dans les images. C'est pourquoi, avant de pouvoir acquérir les séquences des mouvements, il faut enregistrer une séquence vidéo sans l'acteur. En pratique, pendant cette acquisition, nous faisons varier l'éclairage afin de rendre robuste les algorithmes de soustraction de fond aux variations lumineuses naturelles. Cette acquisition doit être répétée s'il survient un changement dans la scène :

- Mouvement d'un objet de le champ de vue d'une caméra,
- Mouvement d'une caméra,
- Changement notable des conditions d'éclairage.

Les séquences de fond permettent de construire un modèle statistique pour chaque pixel. Ce modèle doit être suffisamment robuste pour supporter les variations modérées de lumières et les ombres. De la robustesse du modèle dépendra la qualité des silhouettes qui serviront à la capture de mouvement.

Acquisition des séquences La fréquence d'acquisition des caméras est relativement faible puisqu'elle est de l'ordre de 30 images/seconde. Cette fréquence contraint la vitesse d'exécution des mouvements. En effet des mouvements trop rapides peuvent entraîner l'apparition de flou dans les images si le temps d'exposition des caméras est trop long. Pour palier à ce flou, un bon éclairage de la scène est nécessaire (pour réduire le temps d'exposition des caméras).

De plus, les mouvements doivent être adaptés au nombre de caméras. Au cours de la thèse, j'ai pu tester diverses configurations avec un nombre variable de caméras. A Rennes, nous disposions de quatre caméras disposées en demi-cercle (c.f. illustration 6.2-(a)), tandis que sur la plate-forme GRIMAGE, nous avions six à huit caméras avec différentes configurations se rapprochant de celle illustrée par la figure 6.2-(b).

Les séquences vidéo sont enregistrées au format *raw* ou avec une compression sans pertes. Pour le format non compressé, l'espace disque nécessaire est conséquent : pour une vidéo de dix secondes, l'espace de stockage est

$$\underbrace{10}_{\text{sec.}} \times \underbrace{30}_{\text{img./s.}} \times \underbrace{3}_{\text{R,G,B}} \times \underbrace{780 \times 580}_{\text{taille image}} = 400\text{Mo} \tag{6.1}$$

pour chaque caméra.

Le système VICON Les acquisitions que nous avons effectuées à Rennes ont été menées en parallèle avec un système VICON. Ce dernier nous a permis d'obtenir des données

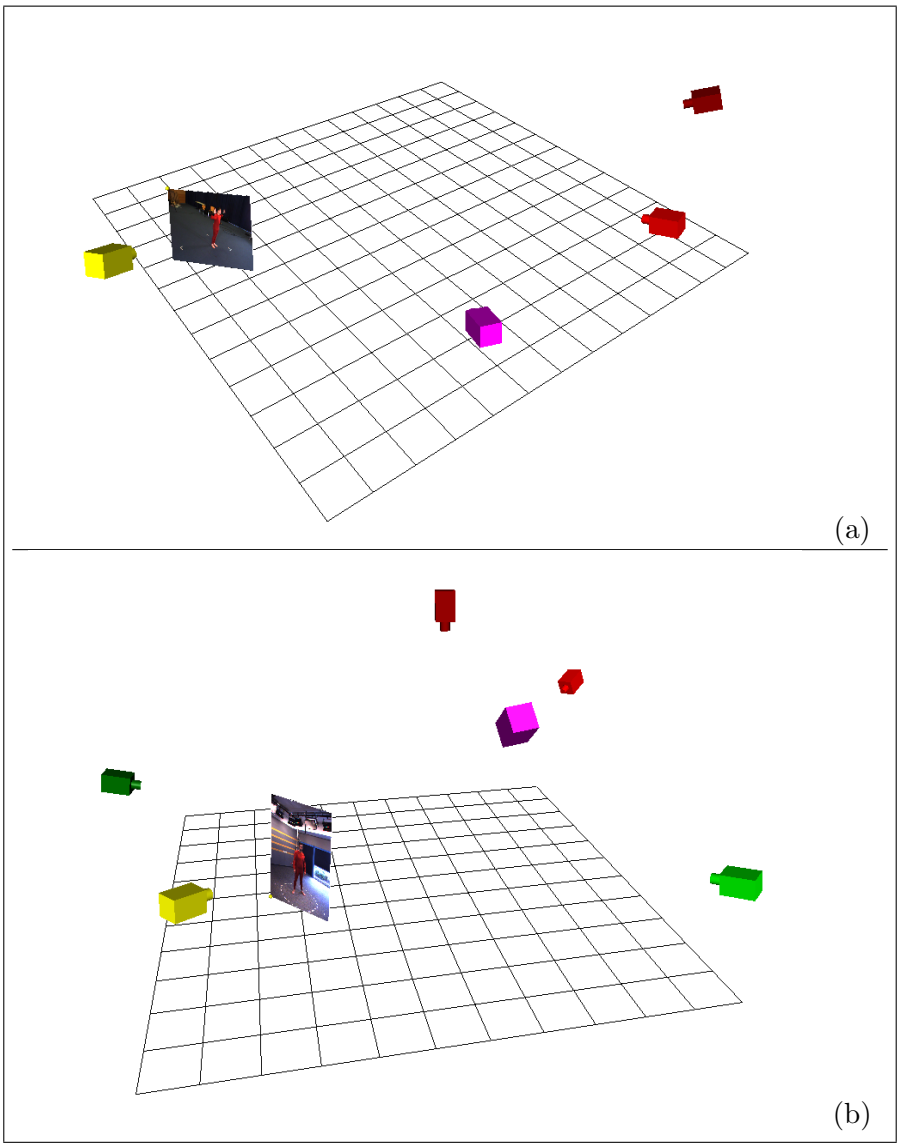


FIG. 6.2: Selon la configuration de caméra utilisée, les mouvements peuvent être plus ou moins complexes. La configuration (a) oblige l'acteur à effectuer des mouvements face aux caméras et quasi-planaires. La seconde configuration (b) autorise n'importe quel mouvement, mais dans un espace restreint.

considérées comme vérité terrain pour évaluer la précision de notre algorithme d'estimation du mouvement.

Nous avons utilisé le système avec huit caméras. Chacune des caméras est dotée d'une unité de traitement permettant d'extraire les marqueurs vus dans les images. Toutes les caméras sont reliées à une unité de synchronisation et de traitement des données. Cette unité a donc la charge d'effectuer la reconstruction *3D* en temps réel des marqueurs. Les caméras sont des systèmes rapides pouvant avoir une fréquence d'acquisition de 120 Hz. Le coût du système est de l'ordre de 300 000 €, ce qui inclut le matériel et les logiciels de calibrage et d'exploitation.

Les traitements Ceux-ci sont effectués selon une procédure naturelle. Dans un premier temps, le système est calibré. Puis les modèles de fond pour chacune des séquences vidéo sont construits. Ensuite, le modèle *3D* de l'acteur est dimensionné. Vient l'estimation du mouvement avec la génération d'un fichier de type BVH (décrit dans l'annexe C). Ce fichier permet d'échanger avec l'UHB les données des mouvements estimés. L'UHB utilise le fichier BVH pour générer un squelette adimensionné ainsi que le mouvement associé, filtré et corrigé, pour pouvoir animer différents modèles graphiques d'acteur. Cette dernière animation est utilisée par ARTEFACTO pour pré-produire des cinématiques de jeux vidéo ludiques.

6.1.3 Les logiciels

Dans la section précédente, nous avons présenté le matériel que nous avons utilisé pour effectuer la capture du mouvement, que ce soit avec les partenaires du projet à Rennes ou au sein de l'INRIA avec la plate-forme GRIMAGE. A l'INRIA, avons implémenté un prototype complet pour le suivi :

- MVCAMERA qui permet la vérification du fonctionnement du système et la calibration des caméras. Deux captures d'écran montrent l'application. 6.3 montre la visualisation des données vidéos acquises. 6.4 est une vue *3D* de la position des caméras avec la trajectoire du bâton de calibration au centre.
- MVBACKGROUND est un outil en ligne de commande permettant d'effectuer l'apprentissage du modèle de l'image pour la soustraction de fond.
- MVACTOR permet de dimensionner le modèle *3D* pour que celui-ci soit correctement adapté à la morphologie de l'acteur (des captures d'écran sont visibles avec la figure **??** du chapitre 5).
- MVPOSER permet d'effectuer l'estimation des paramètres de pose. Les figures 6.5, 6.6 et 6.7 sont des captures d'écran de l'application.

Dans la suite de ce paragraphe, nous présentons rapidement les outils que nous avons utilisés pour implémenter ces logiciels.

Interface Graphique L'ensemble des interfaces a été développé avec QT[1]. Cet environnement nous permet de créer une interface graphique permettant de visualiser

[1] http ://www.trolltech.com

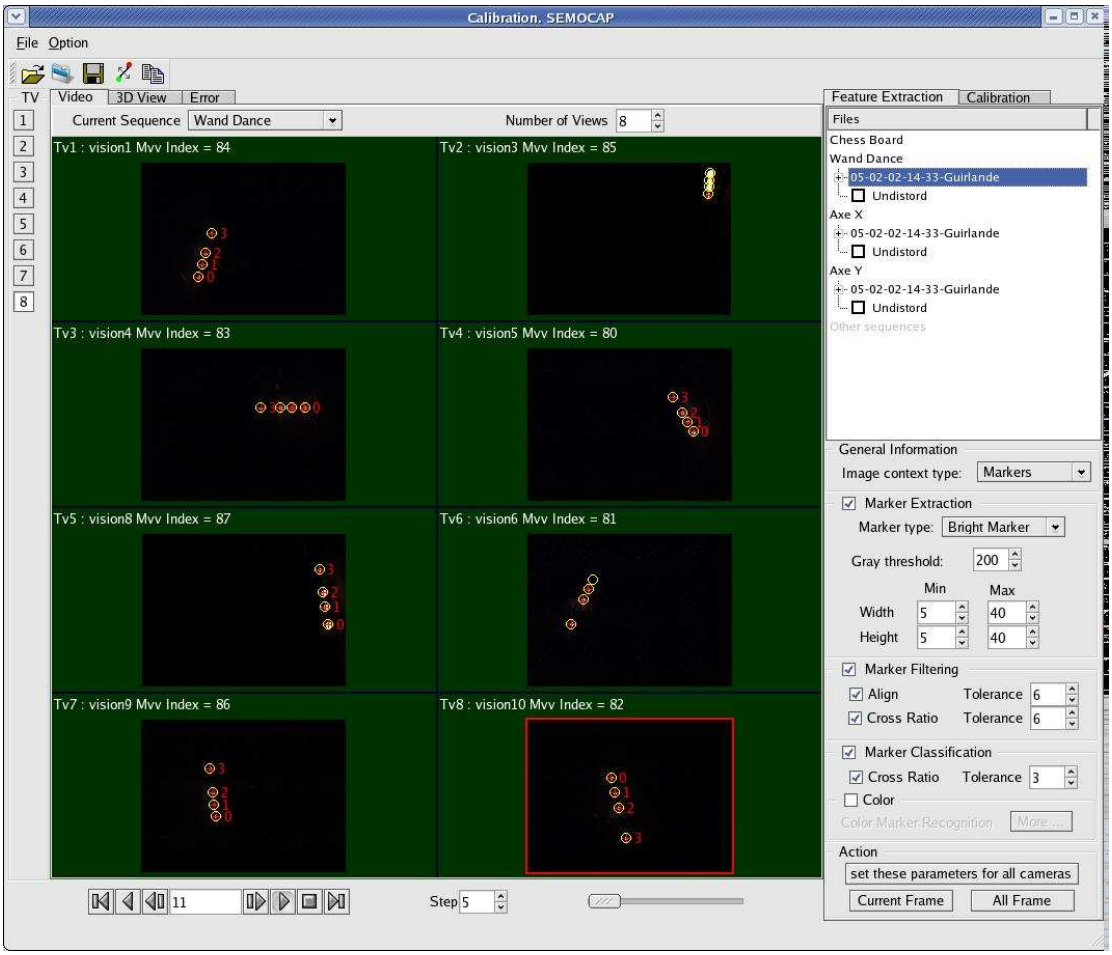


FIG. 6.3: La vue principale de l'interface permet de régler les paramètres pour la calibration et de visualiser les images avec les marqueurs extraits.

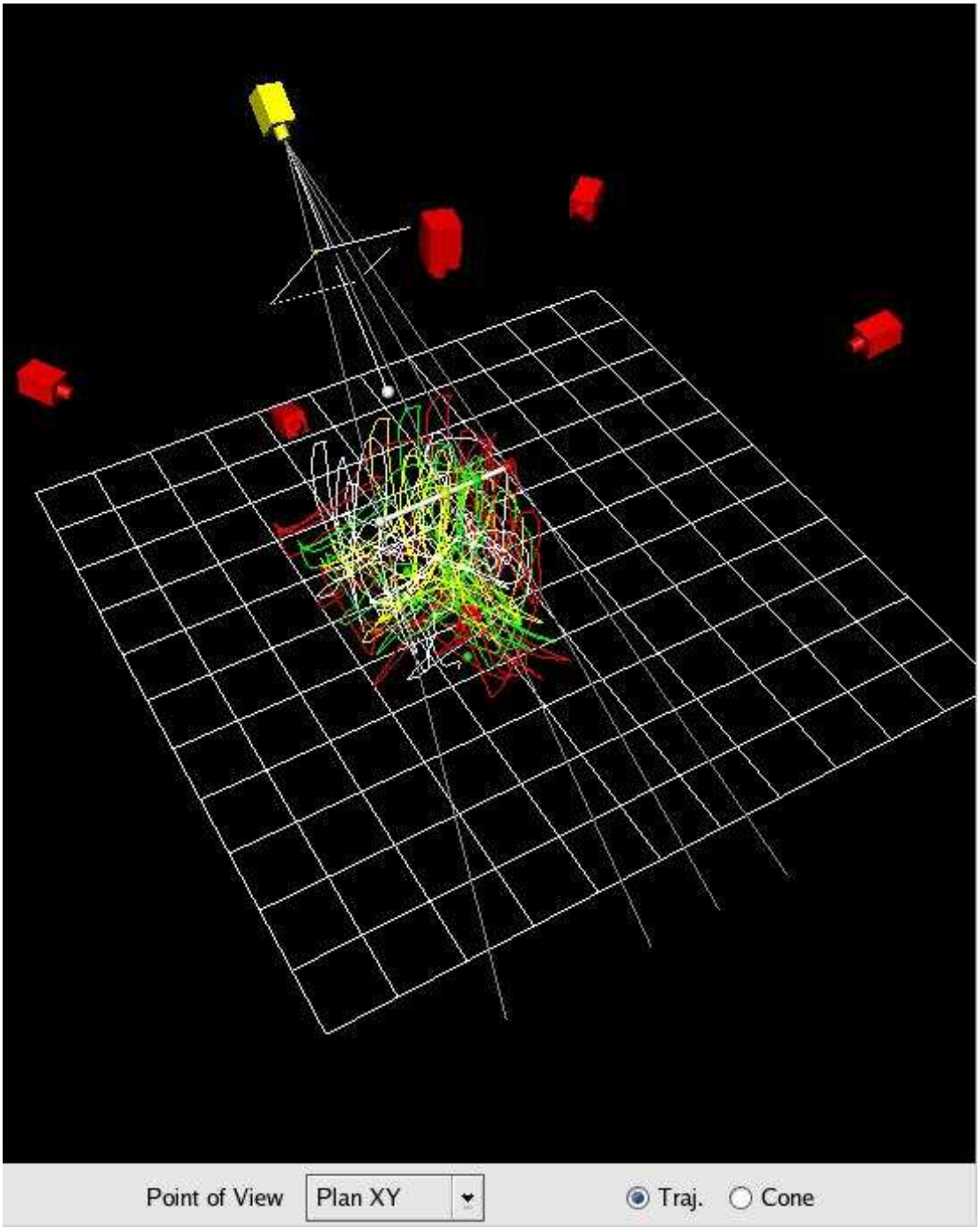


FIG. 6.4: Nous représentons dans la vue *3D* l'ensemble des caméras ainsi que les trajectoires des marqueurs au cours de la séquence de calibration.

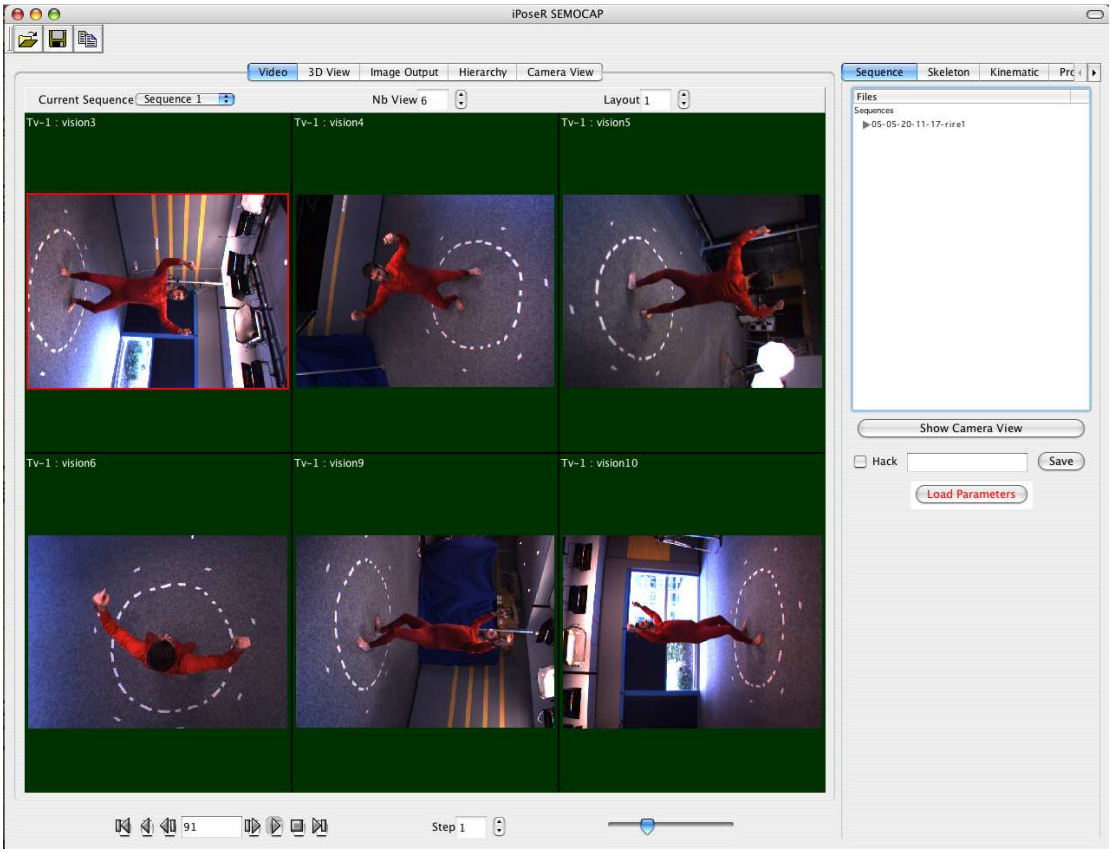


FIG. 6.5: La vue principale de l'interface permet de régler les paramètres pour la calibration et de visualiser les images avec les marqueurs extraits.

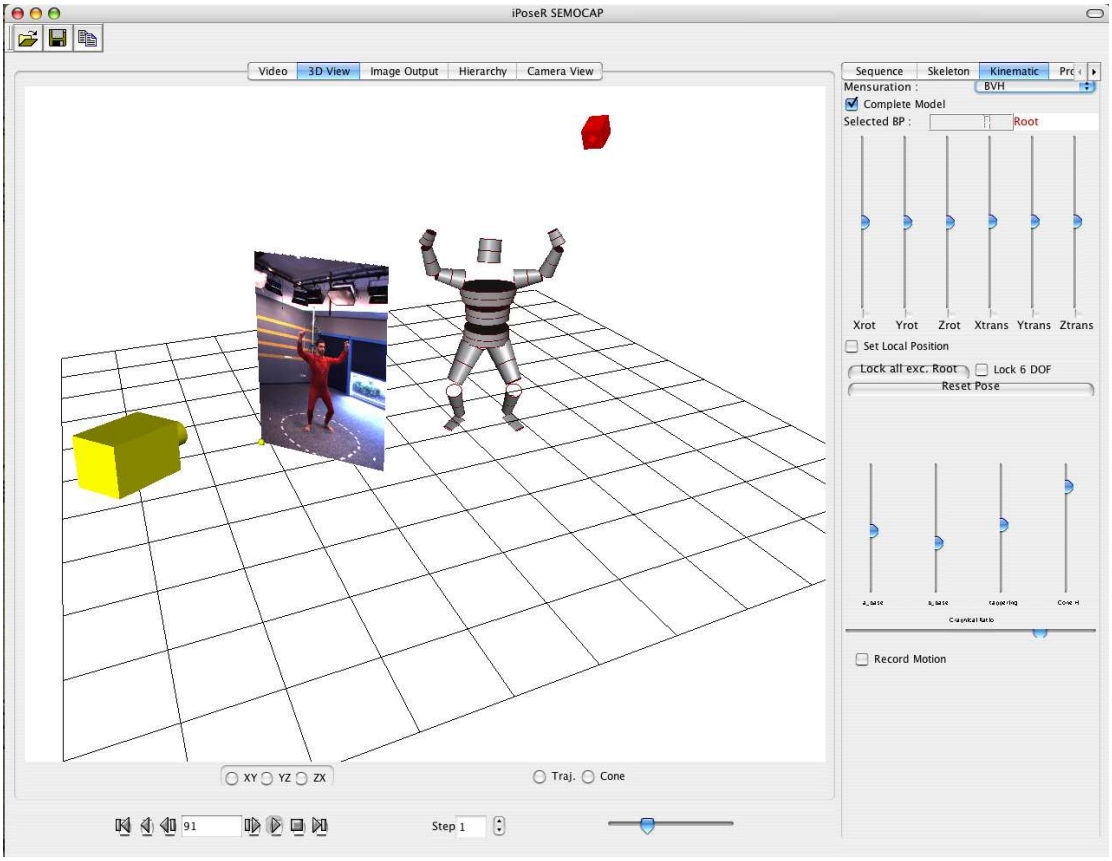


FIG. 6.6: La vue OpenGL de l'application permet de visualiser en *3D* et donc plus facilement l'évolution du modèle lors de l'estimation du mouvement. A droite nous pouvons apercevoir des curseurs permettant de rectifier la pose de l'acteur si nécessaire

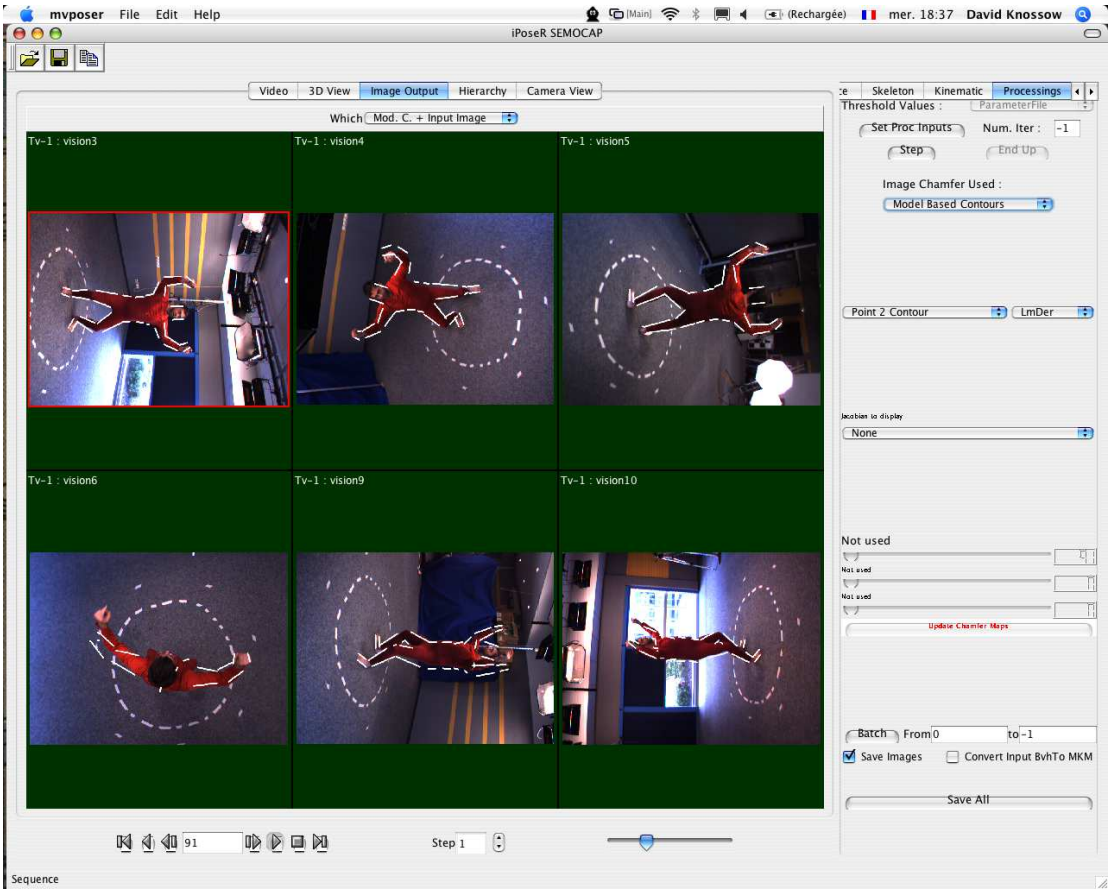


FIG. 6.7: Cette vue, différente de la vue principale, permet d'afficher divers résultats comme les images avec les silhouettes, les cartes de chanfrein, le modèle *3D* projeté dans les images *etc*.

l'ensemble des résultats mais aussi d'interagir avec l'ensemble des algorithmes que nous avons implémentés (notamment le réglage dynamique des paramètres, comme par exemple lors de la détection de contours standard). Avec cette interface nous pouvons afficher aussi bien des données *2D* comme des images que du contenu *3D* comme le modèle *3D* de l'acteur ou les caméras.

Librairies Pour implémenter l'ensemble des algorithmes, nous avons utilisé essentiellement deux librairies : OpenCV et minpack. La première librairie permet d'effectuer les traitements images standards comme la détection de contours (Canny), la transformée en distance, les conversions colorimétriques des images, *etc*. La seconde librairie est une implémentation efficace en Fortran d'algorithmes d'optimisation. Nous l'avons utilisé pour effectuer l'estimation des paramètres (algorithme de Levenberg-Marquardt).

Nous avons aussi utilisé une implémentation du filtrage anisotropique gaussien proposé dans [144]. Cette implémentation efficace du filtrage nous permet de calculer les cartes de contours utilisant le modèle *3D*.

Enfin, la librairie OpenGL nous a permis de calculer les cartes de visibilité du modèle *3D*, mais aussi les diagrammes de Voronoï comme décrit au chapitre 5. Ces calculs sont en réalité effectué *off-screen* à l'aide des *pbuffers*.

6.2 Résultats

Nous allons présenter dans cette section différents résultats obtenus sur diverses séquences vidéos. Nous allons présenter les résultats avec les différentes méthodes que nous avons évoquées au cours de la thèse : utilisation des contours (extraits avec un détecteur standard ou alors utilisant les contours) et utilisation de la couleur.

Dans une première partie, nous présentons des résultats sur des images synthétiques permettant de mettre en avant les performances de la méthode de suivi utilisant les contours. Nous présentons des résultats de suivi du mouvement pour diverses séquences de mouvement. Tout au long de la thèse, nous avons fait évoluer les diverses techniques de suivi du mouvement. Nous montrerons les résultats des diverses techniques sur différentes séquences réelles acquises aussi bien sur la plate-forme GRIMAGE qu'à Rennes dans le cadre du projet SEMOCAP.

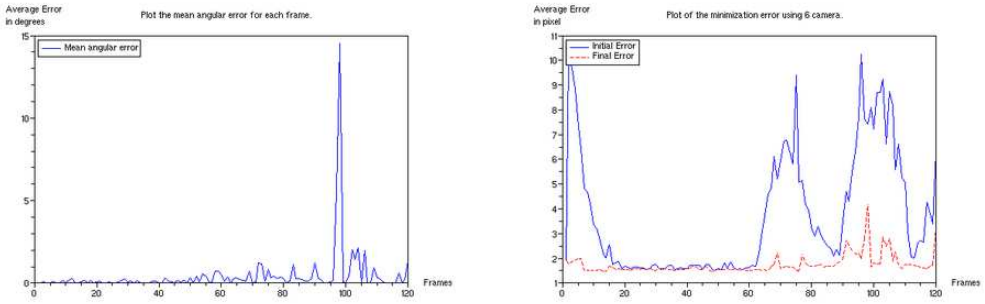
6.2.1 Résultats synthétiques

Les données synthétiques ont été créées à partir d'une estimation du mouvement sur une séquence vidéo réelle. Il s'agit donc d'un mouvement estimé que nous utilisons comme vérité terrain. Ce mouvement est donc réaliste et permet d'évaluer les performances des algorithmes sur des données idéales. Un modèle *3D* est animé à partir de cette estimation. Ce modèle est projeté dans des images afin de créer les silhouettes et les contours de ce modèle. Nous utilisons les images générées comme données d'entrée de l'algorithme.

La figure 6.9 montre le modèle *3D* de référence dans différentes postures (premières lignes) ainsi que la pose estimée (secondes lignes). Les figures 6.8, 6.10, et 6.11 montrent une comparaison de l'estimation des paramètres angulaires avec la vérité terrain. Le graphique 6.8-(a) représente l'erreur moyenne exprimée en degrés entre les paramètres de pose de la vérité terrain et ceux estimés. Nous pouvons constater que l'erreur moyenne est de moins de deux degrés excepté à l'image 98 (où nous avons une erreur de l'ordre de 15 degrés). Cette dernière erreur est liée à la mauvaise estimation du mouvement des mains. Cependant, l'algorithme retrouve correctement les paramètres dans la suite du suivi. Le graphique 6.8-(b) représente l'erreur initiale et l'erreur après minimisation en utilisant la méthode associée aux contours avec la distance de Hausdorff (chapitre 5 section 5.2.2). L'erreur angulaire à l'image 98 se traduit par une erreur moyenne de l'ordre de 4 à 5 pixels.

Les figures 6.10 et 6.11 comparent de manière détaillée les trajectoires angulaires estimées avec les trajectoires de la vérité terrain. Nous présentons la comparaison pour les deux d.d.l. du coude et deux des trois degrés de liberté (d.d.l.) de l'épaule.

En conclusion, l'algorithme se comporte bien sur des données synthétiques.



(a) (b)

FIG. 6.8: (a) – Erreur moyenne entre le mouvement simulé et le mouvement estimé, exprimée en degrés. (b) – Erreur moyenne exprimée en pixels avant minimisation (courbe la plus haute) et après minimisation (courbe basse).

6.2.2 Suivi de mouvements sur des séquences réelles

Dans cette section, nous allons présenter différents résultats acquis avec différentes configurations de caméras.

Pour effectuer les acquisitions, nous avons utilisé deux systèmes, l'un avec 6 caméras à l'INRIA, et l'autre avec 4 caméras à Rennes. Les deux systèmes sont équivalents. Les différences résident dans les optiques des caméras et la flexibilité du système. A Rennes, nous avons opté pour un système démontable aisément. Les caméras sont donc montées sur pieds et déplaçables selon la configuration voulue. Sur la plate-forme GRIMAGE, les caméras sont montées sur un portique.

La différence concernant les optiques des caméras est importante. A Rennes, les optiques choisies n'ont pas permis de faire des acquisitions rapides pour diverses raisons techniques : pas de focus sur les objectifs et une ouverture très petite. Nous avons donc dû régler un temps d'exposition assez long pour avoir assez de luminosité. En contre partie, le mouvement devait être plus lent sous peine de flou de bougé dans les images. Mais cette configuration nous a permis un volume d'acquisition assez élevé (c.f. illustration 6.12). La synchronisation précise (de l'ordre de 10^{-6} seconde) alliée à l'utilisation d'un *shutter* rapide pour les caméras nous a permis de traiter des mouvements rapides lors des acquisitions à l'INRIA. La configuration des caméras est illustrée avec la figure 6.13. Nous pouvons voir que le volume d'acquisition est plus faible qu'avec la configuration de Rennes.

Pour les différentes acquisitions, nous avons eu différents acteurs : Erwan, Ben et Stéphane. Chacun de ces acteurs a une morphologie différente, nous avons donc adapté le modèle *3D* générique à chacun de ces acteurs. Nous montrons les modèles sur la figure 6.14 avec des poses similaires.

Dans la suite de ce paragraphe, nous présentons divers résultats du suivi de mouvement.

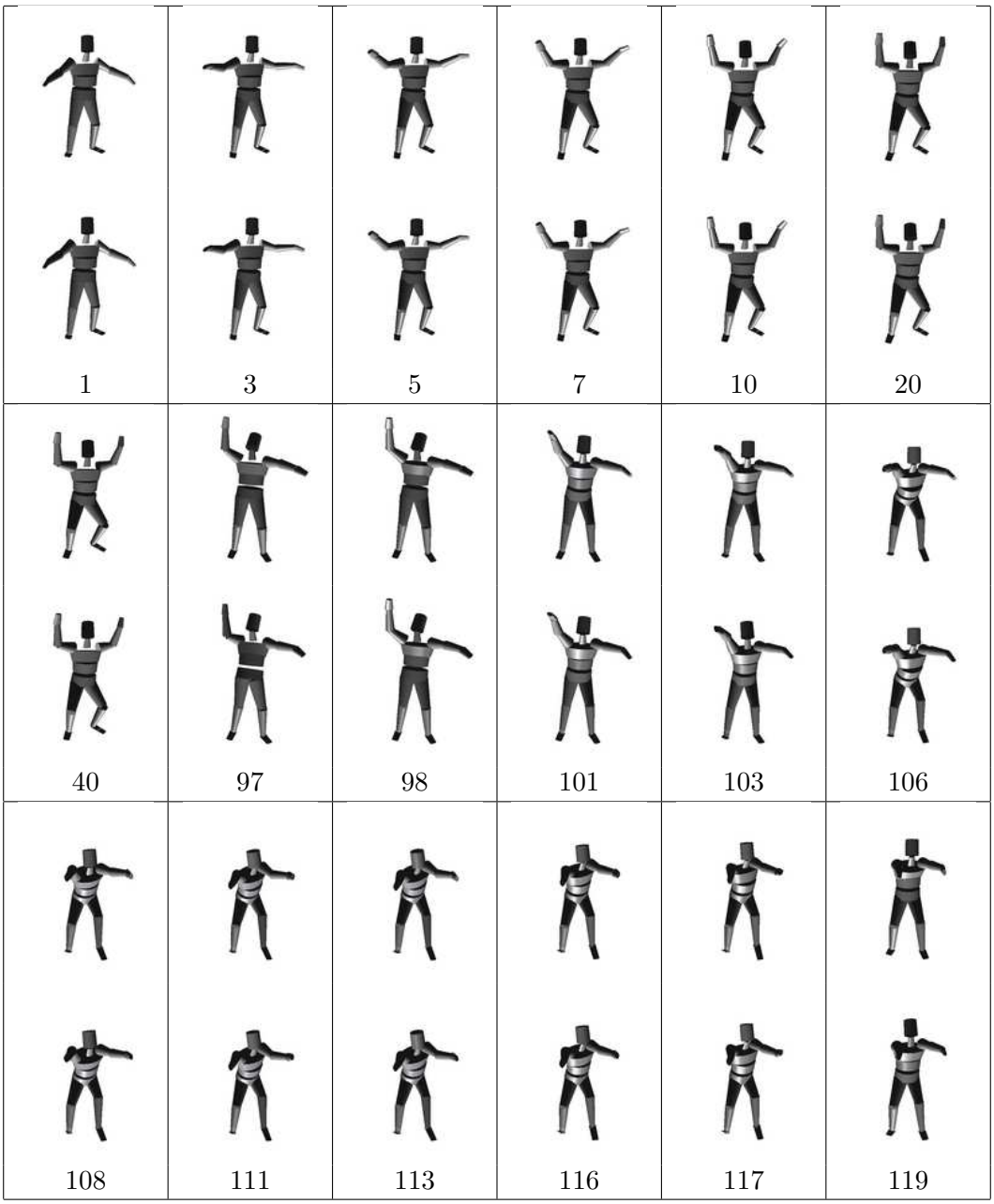


FIG. 6.9: Une séquence de mouvements simulés est prise comme vérité terrain (premières lignes). A partir de cette séquence, nous générons les contours et les silhouettes pour effectuer le suivi. La pose est alors estimée (secondes lignes).

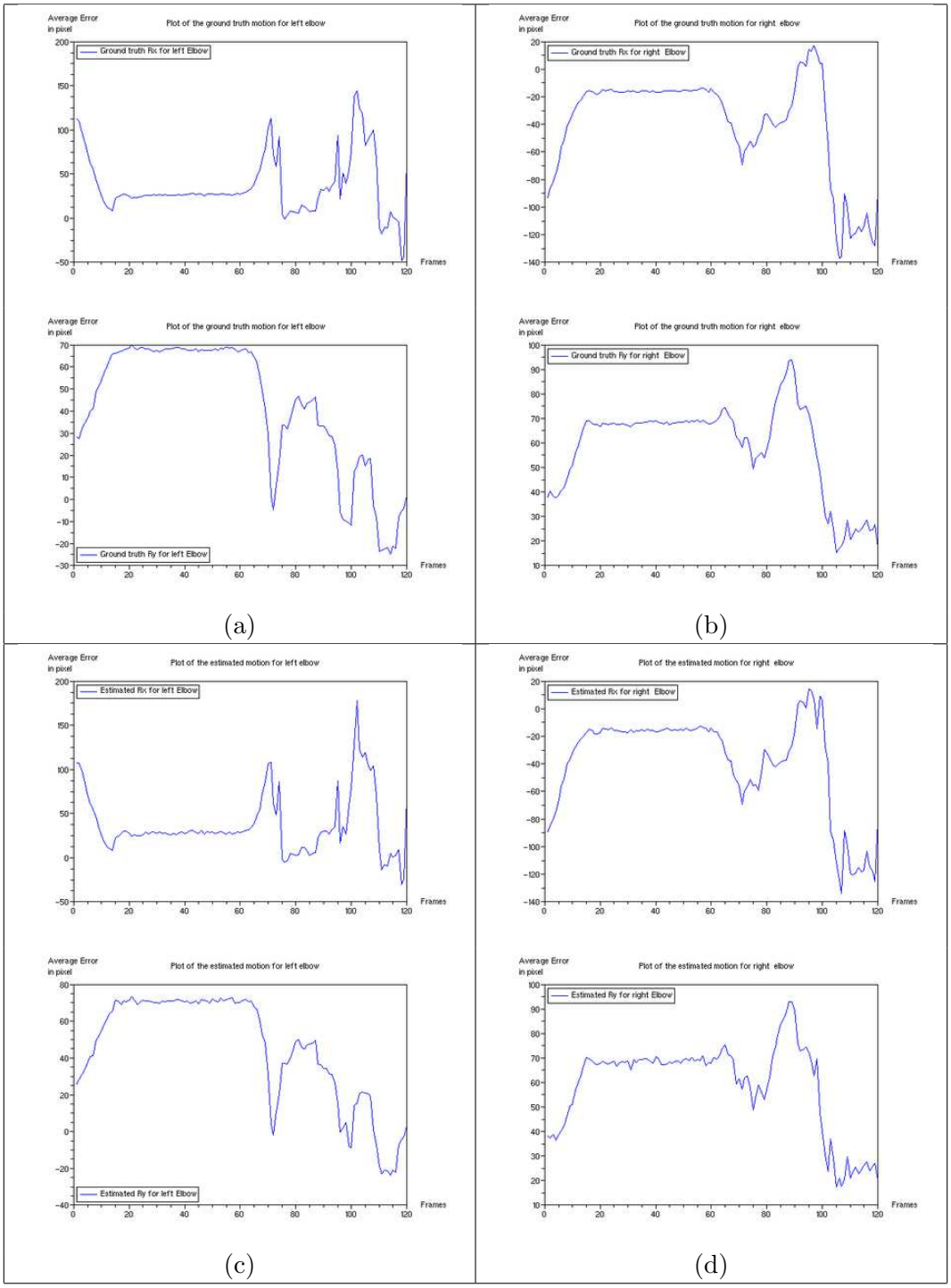


FIG. 6.10: Vérité terrain (a,b) et estimation des trajectoires angulaires des coudes gauche et droit (c, d).

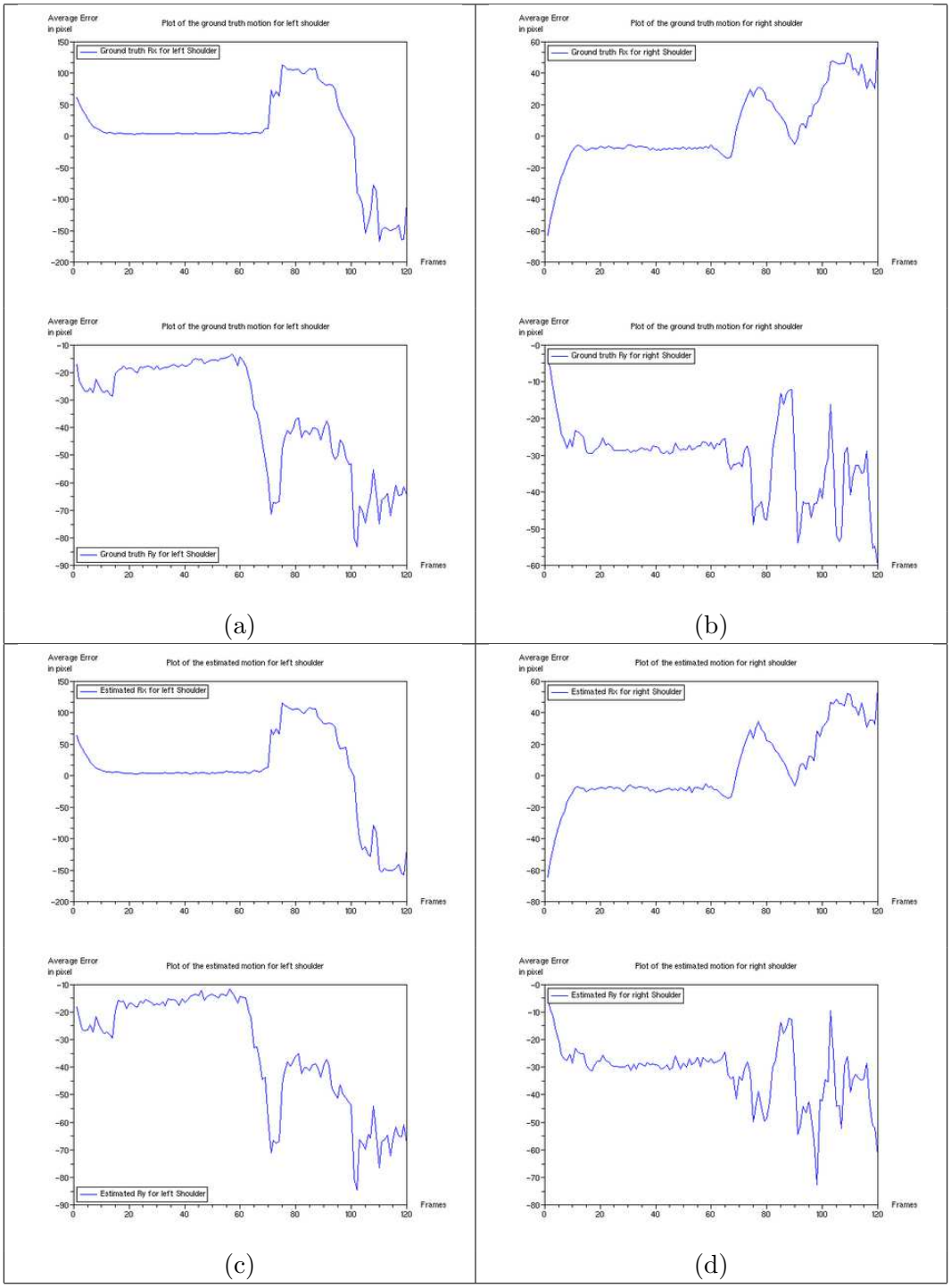


FIG. 6.11: Vérité terrain (a,b) et estimation des trajectoires angulaires des épaules gauche et droit (c, d).

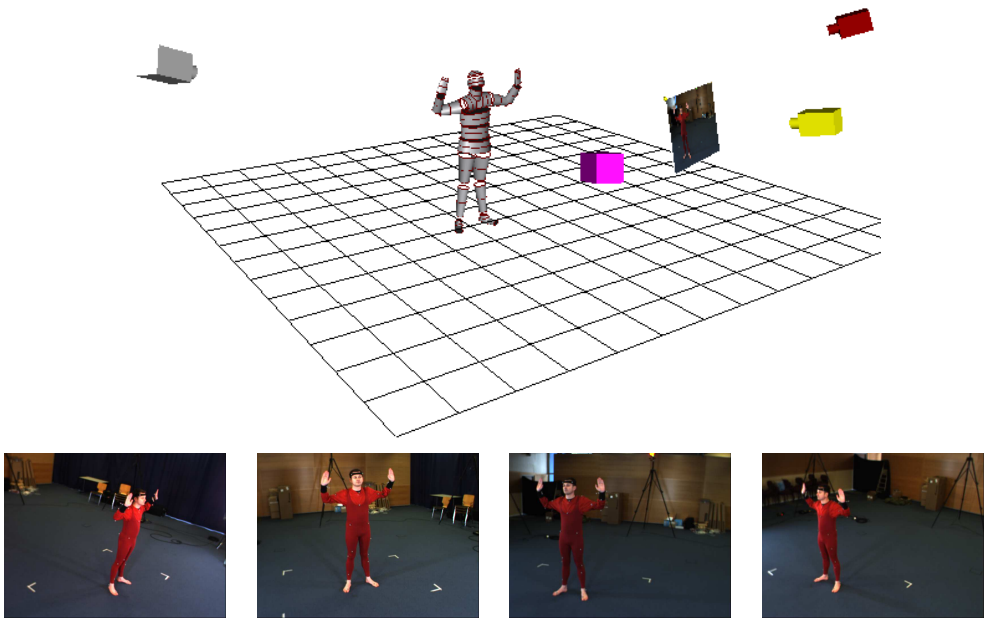


FIG. 6.12: Configuration des caméras utilisée lors de nos acquisitions à Rennes. Nous représentons quatre points de vue de l'acteur Stéphane.

Les premiers résultats que nous avons obtenus avec l'algorithme utilisant les silhouettes sont présentés sur la figure 6.15. Il s'agit d'une séquence de mouvement, de 800 images, au cours de laquelle Ben effectue un mouvement des bras. Nous avons ensuite amélioré la stabilité de la minimisation en rajoutant la détection de contours standard (Canny sur les images en niveau de gris). Nous avons estimé le mouvement sur une séquence rapide de mouvement (30 images pour le mouvement complet) qui est illustré avec la figure 6.16. Sur cette séquence, nous avons utilisé un modèle d'acteur ajusté à la main. Nous pouvons voir que l'adaptation du modèle n'est pas parfaite avec notamment le bassin qui n'est pas correctement ajusté pour prendre en compte l'extension des jambes lorsque le pied est levé. Ce n'est que vers le milieu de la seconde année de thèse que nous avons obtenu les premiers résultats du dimensionnement de l'acteur de manière semi-automatique. Les résultats suivants utilisent donc le modèle *3D* dimensionné avec cette méthode. La figure 6.17 représente quelques images d'une séquence qui en compte 250. La complexité du suivi est liée au fait que l'ensemble des degrés de liberté entrent en jeu. Nous avons utilisé la technique de détection de contours utilisant le modèle *3D*. L'intérêt de cette technique n'est pas évident pour la séquence présentée. Nous montrons l'avantage de cette technique avec la comparaison donnée sur la figure 6.18. Alors que la technique utilisant les silhouettes échoue à suivre les mains, la technique utilisant les contours orientés permet un suivi correct.

Avec les figures 6.19-(a), 6.20-(a), 6.21-(a) nous montrons à gauche les résultats de l'estimation du mouvement appliqués à un avatar tandis qu'à droite il s'agit du

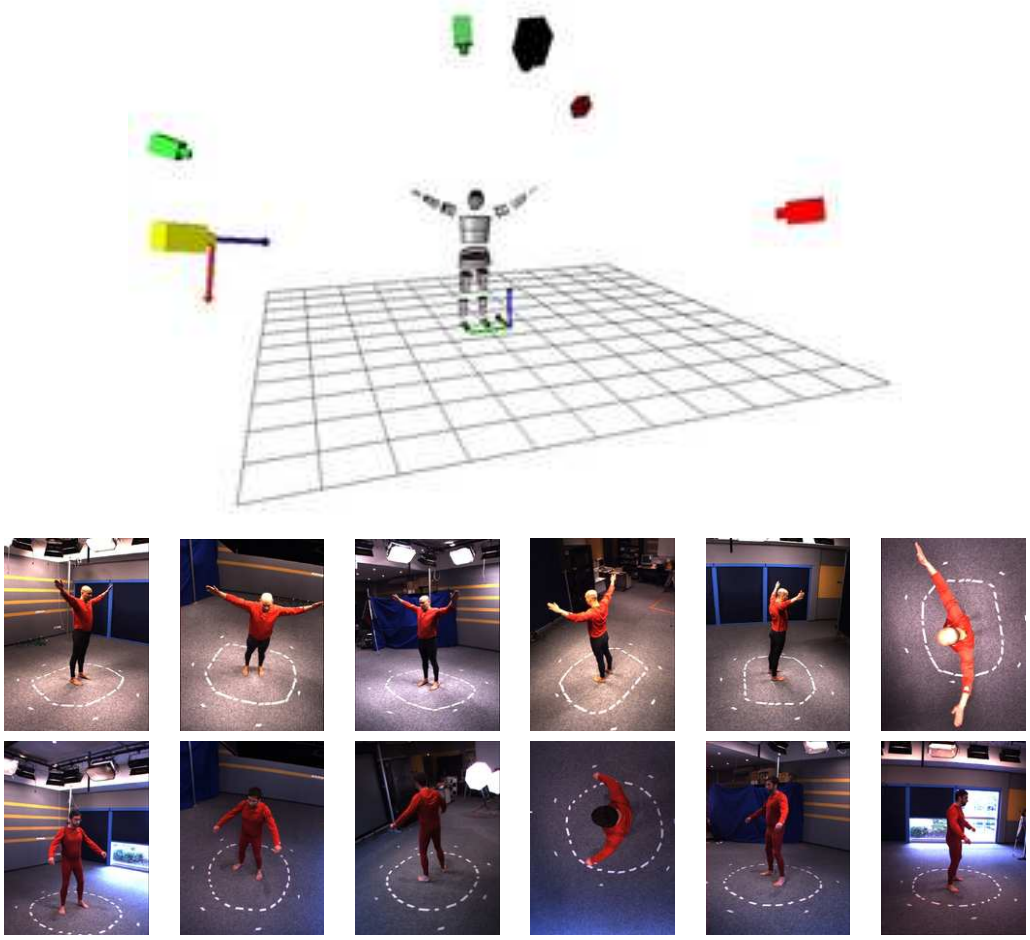


FIG. 6.13: Configuration des caméras utilisée lors de nos acquisitions à l'INRIA. Nous représentons six points de vue de l'acteur Ben et six points de vue de Erwan.

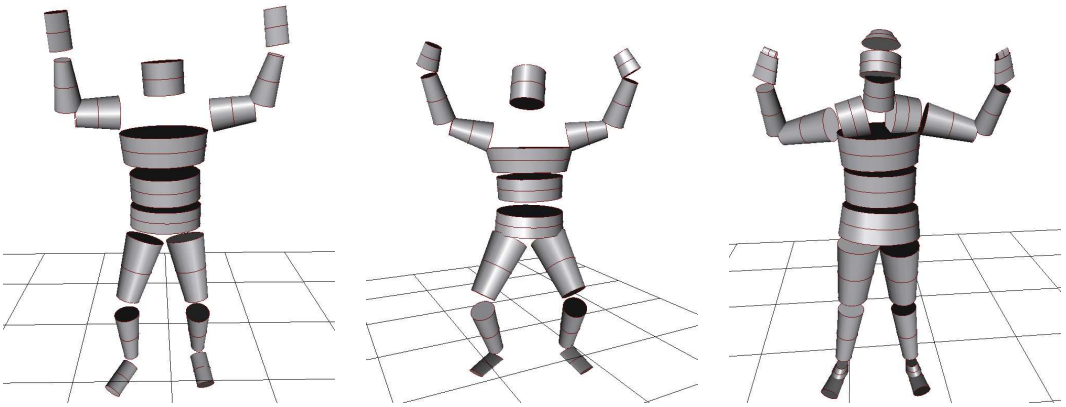
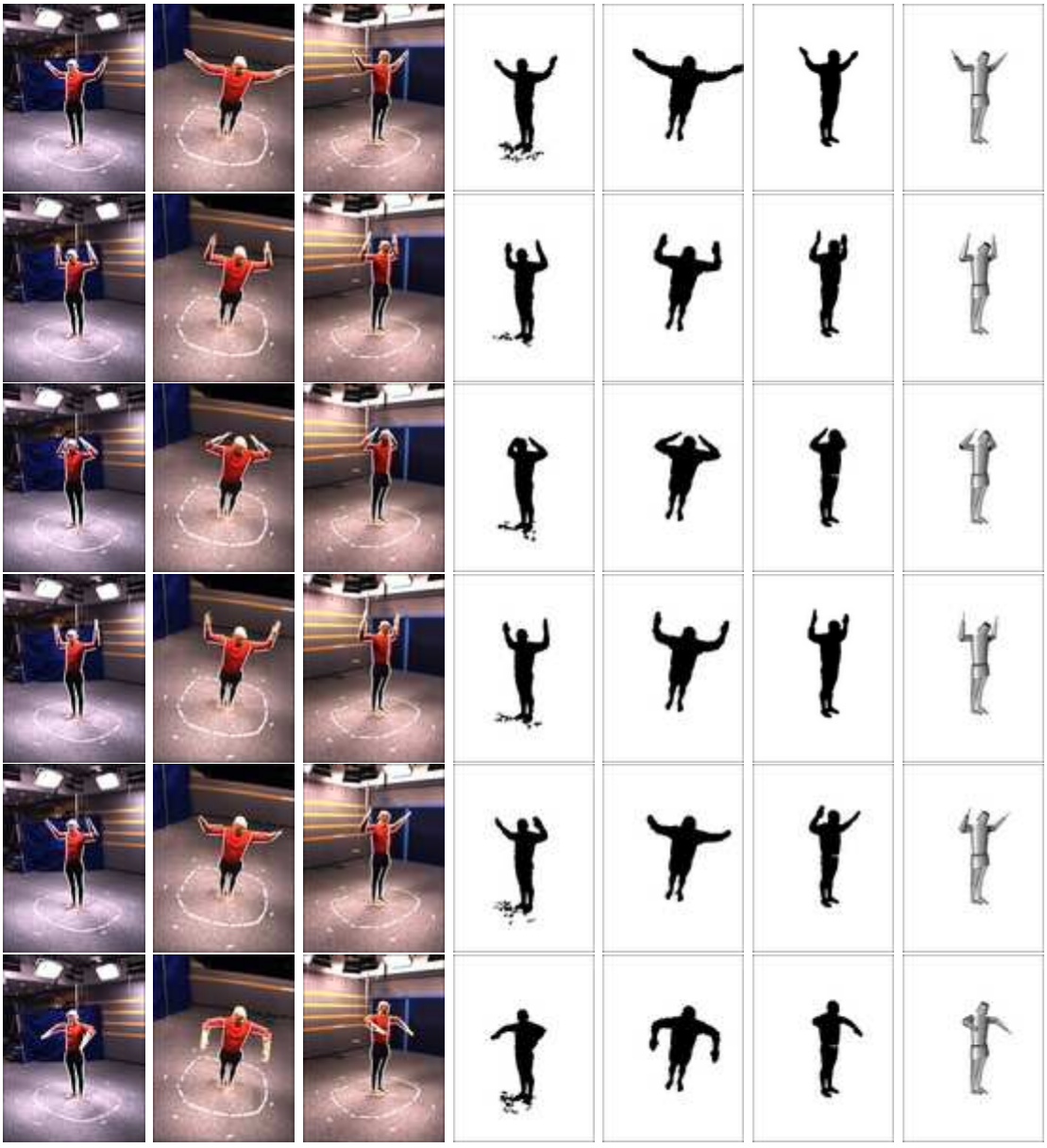


FIG. 6.14: Les modèles d'acteur de Ben, Erwan et Stéphane. Nous pouvons voir que le modèle de Stéphane est un modèle complet tandis que les deux autres sont des modèles simplifiés.

même mouvement corrigé. Ces résultats sont obtenus grâce aux travaux de l'UHB sur le *retargetting* du mouvement. Il s'agit, à partir de données de capture de mouvement, de générer un mouvement adimenssionné applicable à n'importe quel modèle d'acteur humain ([91]).

Les figures 6.19-(b), 6.20-(b), 6.21-(b) comparent les résultats du processus complet de SEMOCAP avec les données brutes de VICON. Nous pouvons voir que l'utilisation des résultats de capture du mouvement donnés par un système à marqueurs (ici le VICON) sans traitement a posteriori ne permet pas d'animer un personnage virtuel correctement.



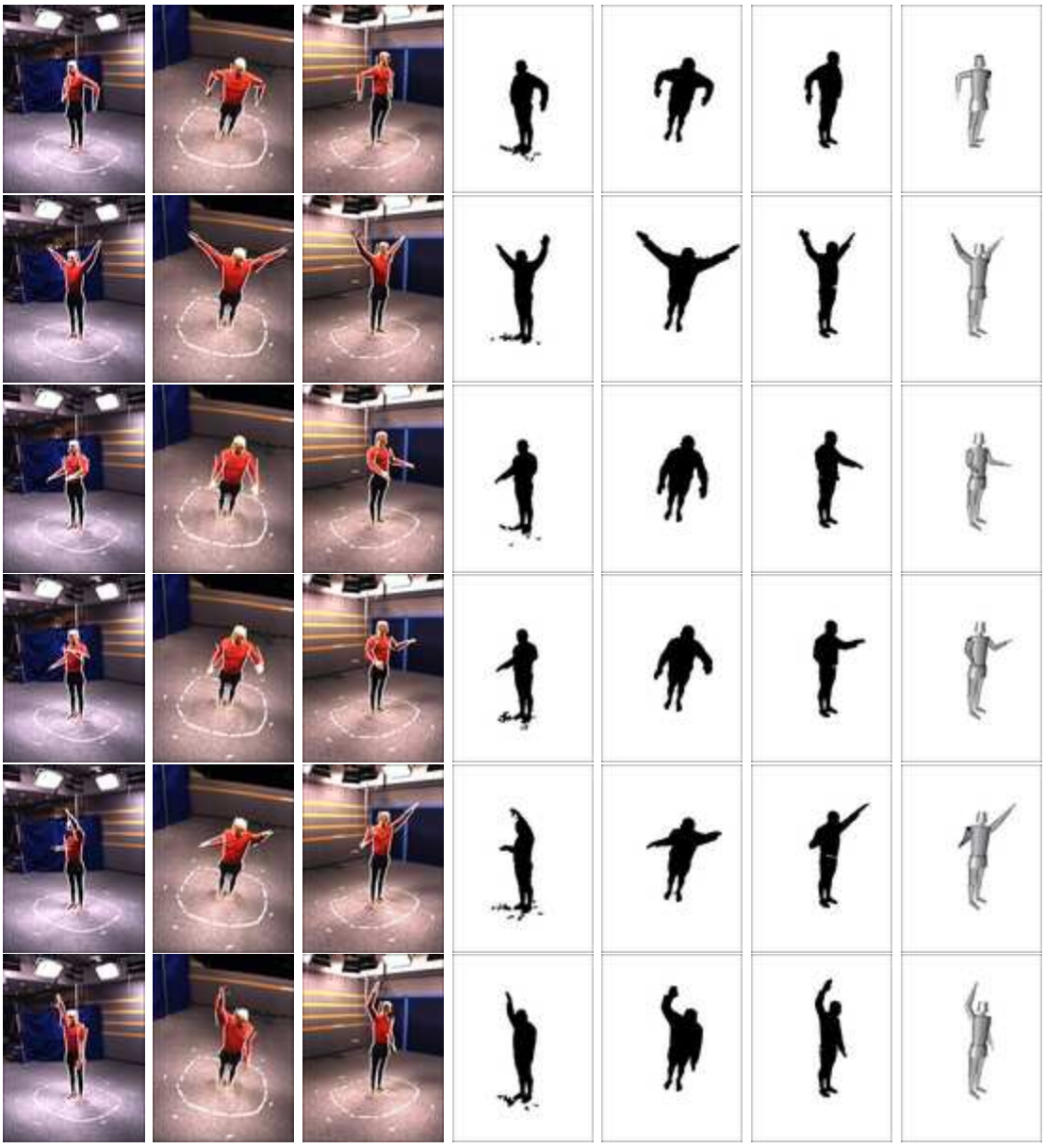


FIG. 6.15: Nous présentons le résultat du suivi du mouvement de gym effectué par Ben qui compte en tout 800 images. En ligne, nous présentons trois points de vue sur les six de la séquence, les silhouettes extraites de chacun de ces points de vue et enfin le résultat de l'estimation de la pose.

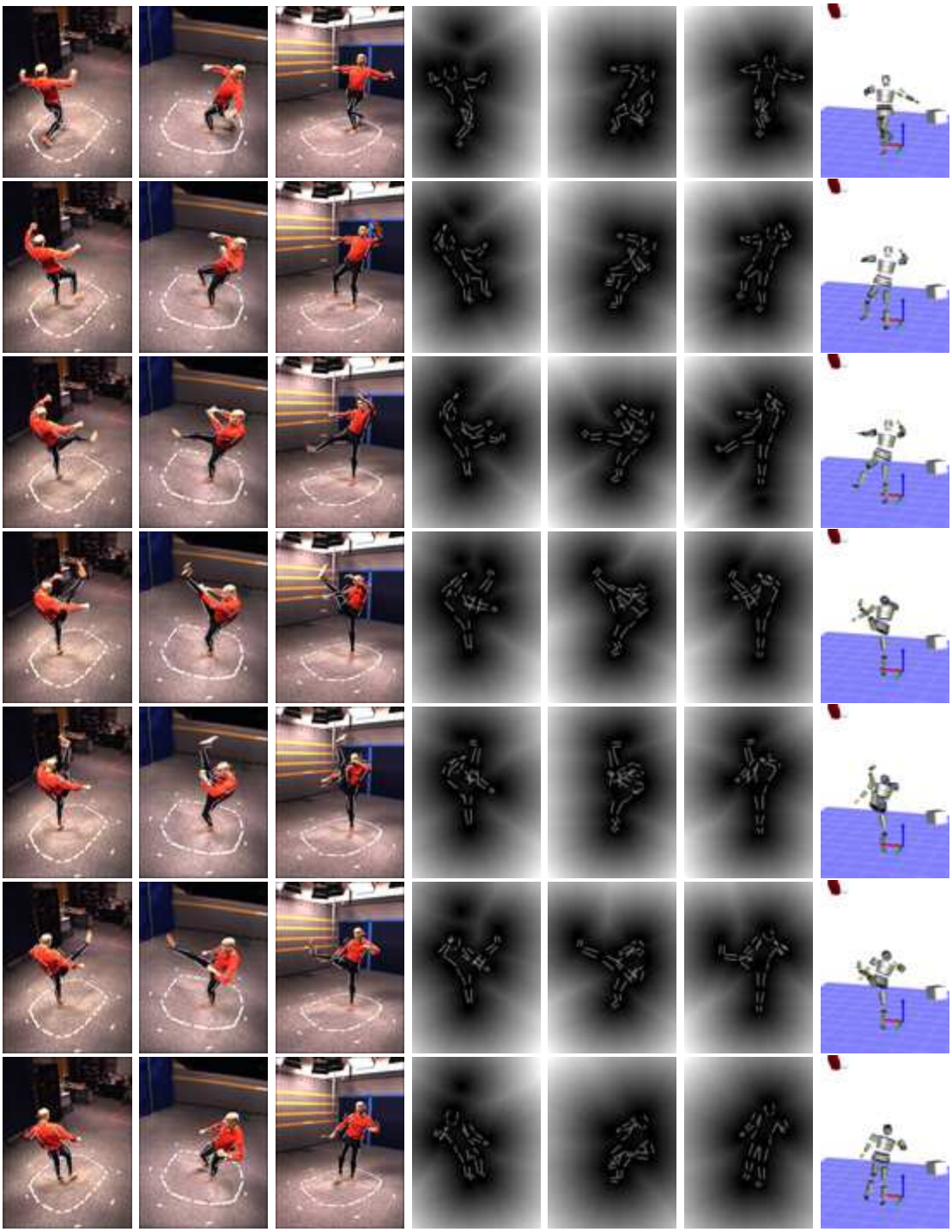


FIG. 6.16: Nous présentons les résultats de l'estimation du mouvement sur une séquence de mouvement très rapide : il s'agit d'un coup de pied effectué par Ben en environ une seconde soit 30 images. Nous utilisons l'algorithme avec les silhouettes et les contours standards pour effectuer l'estimation du mouvement. En ligne, nous présentons trois points de vue des six utilisés, puis les distances de chanfrein calculées avec le modèle *3D* projeté sur celle-ci et enfin le résultat de l'estimation.

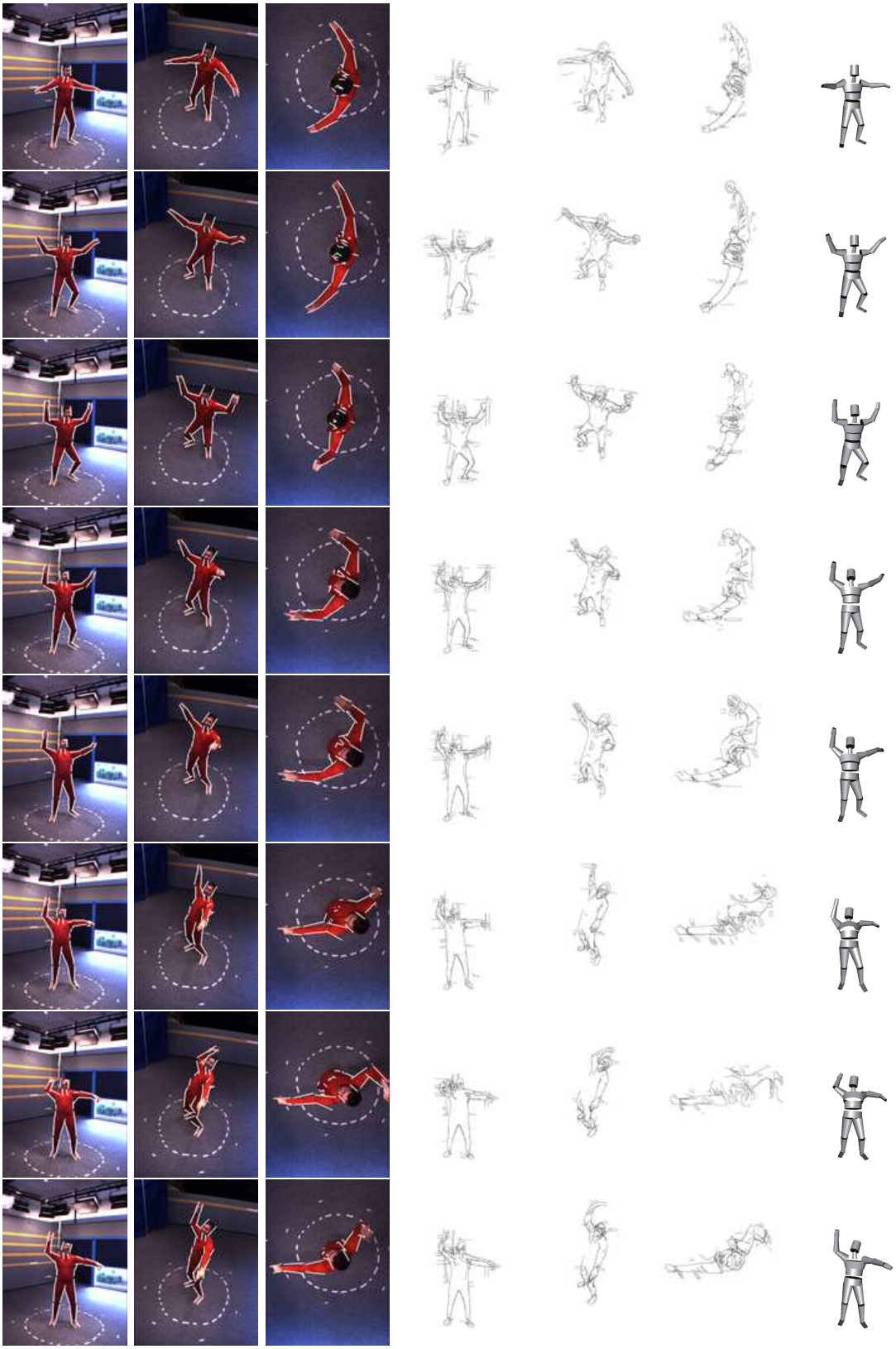


FIG. 6.17: Nous présentons le résultat du suivi du mouvement de rire effectué par Erwan. En ligne, les trois premières images représentent trois des six points de vue que nous utilisons, puis les trois images suivantes représentent les contours extraits à l'aide de la méthode orientée contours, enfin la dernière image est le résultat de l'estimation du mouvement.

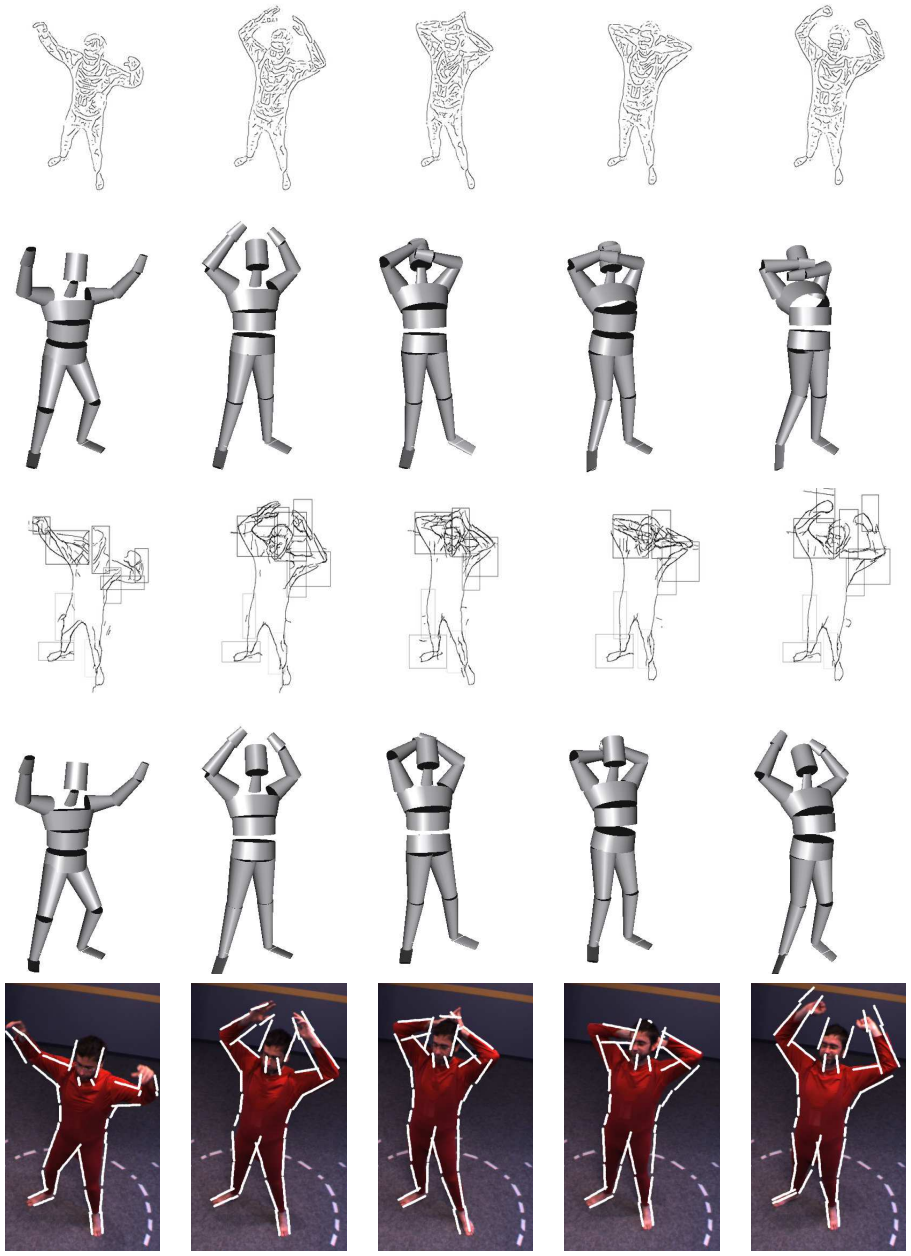


FIG. 6.18: Suivi du mouvement comparé entre la méthode utilisant les contours extraits à l'aide d'un détecteur standard de Canny et la méthode utilisant les contours extraits avec le filtre anisotropique gaussien. La méthode de détection de contours standard est illustrée sur le ligne (1). La pose estimée à l'aide de cette méthode est illustrée sur la ligne (2). Le suivi du mouvement échoue : les bras ne sont pas correctement estimés. Le filtrage anisotropique gaussien (ligne (3)) permet de suivre correctement le mouvement humain. Les résultats sont donnés sur la ligne (4). Enfin, la dernière ligne est la projection du modèle *3D* sur les images.

(a) (b)

FIG. 6.19: (a) - A gauche, le mouvement que nous estimons est appliqué à un avatar, tandis qu'à droite le mouvement est corrigé à l'aide de l'application MKM. La seule différence notable est au niveau de la tête. (b) - Nous comparons le résultat de SEMOCAP avec celui donné par VICON. Les résultats de VICON sont moins bons car ils nécessitent des post-traitements pour être adaptés à la morphologie de l'avatar.

(a) (b)

FIG. 6.20: (a) - A gauche, le mouvement que nous estimons est appliqué à un avatar, tandis qu'à droite le mouvement est corrigé à l'aide de l'application MKM. Nous pouvons voir qu'au niveau des mains, le contact n'est pas correct sur l'estimation et est corrigé. De plus la cuisse est mal orientée et est donc corrigée.(b) - Nous comparons le résultat de SEMOCAP avec celui donné par VICON. Nous pouvons voir que les mains ne se touchent pas avec les résultats VICON.

(a) (b)

FIG. 6.21: (a) - A gauche, le mouvement que nous estimons est appliqué à un avatar, tandis qu'à droite le mouvement est corrigé à l'aide de l'application MKM. On peut voir que les pieds ne sont pas sur le sol sur l'estimation, une contrainte de contact entre les pieds et le sol est imposée par MKM. (b) - Nous comparons le résultat de SEMOCAP avec celui donné par VICON.

6.3 Discussions

6.3.1 Les données d'entrée

Nous analysons ici les conditions dans lesquelles les résultats ont été obtenus et plus précisément les conditions pour que l'estimation du mouvement et plus particulièrement la minimisation réussisse. Le raisonnement que nous allons proposé ici démarre en supposant que nous avons une seule caméra. L'extension pour plusieurs caméras sera faite au cinquième paragraphe.

Pour que la minimisation soit correcte, une condition est nécessaire : la Hessienne de la fonction d'erreur doit être de rang plein et donc inversible. Or l'algorithme de Levenberg-Marquardt fait une approximation de cette matrice avec $\mathbf{H} = \mathbf{J}_{\boldsymbol{v}_E}^{\top}\mathbf{J}_{\boldsymbol{v}_E}$, où $\mathbf{J}_{\boldsymbol{v}_E}$ est la matrice Jacobienne définie au chapitre 5 section 5.3.2. Elle est de taille $m \times p$, où m est le nombre de points de contours (extrémaux) total (vus par une seule caméra) et p le nombre de paramètres à estimer. Pour que la Hessienne soit bien conditionnée et inversible, nous devons avoir $m \geq p$ et donc au moins p lignes indépendantes.

Chaque point du contour extrémal est représenté par une ligne dans la matrice Jacobienne, donc nous devons avoir un minimum de n points indépendants pour que la Hessienne soit inversible. Si chacune des parties du corps était vue comme un objet rigide doté de six degrés de libertés (d.d.l.), trois points de l'objet vus avec une seule caméra seraient suffisant pour déterminer sa pose. Donc, si nous connaissons les assignations de chaque point du contour modèle avec chaque point du contour image, seuls trois points par partie du corps sont utiles pour estimer la pose de chacune d'elles. Mais la position du point le long du contour est indéterminée (nous n'avons pas de correspondance de points entre les contours observés et le contours du modèle). Six points seraient donc nécessaires pour chaque primitive.

Le cas de la chaîne cinématique est d'autant plus complexe que le nombre de d.d.l. est

de $6+m$ pour la partie extrémale de la chaîne (6 d.d.l. pour la racine et m d.d.l. en rotation). Cependant les différentes parties de la chaîne ne sont en réalité pas indépendantes, ce qui contraint l'estimation de la pose de la partie extrémale (et bien sûr de toutes les autres). Pour estimer l'ensemble de ces d.d.l., il est donc raisonnable de répartir les points sur toutes les parties de la chaîne. Le modèle cinématique du corps humain que nous utilisons est composé de cinq sous chaînes qui partagent une racine commune (c.f. section 4.1), vingt et un segments et quarante quatre d.d.l. (dont six de mouvement libre et trente huit de degrés en rotation). En théorie, avec trois points en moyenne par partie du corps, nous avons suffisamment de données pour estimer l'ensemble des degrés de liberté. En réalité, nous en observons beaucoup plus (environ dix par segments).

Cependant, en pratique, il y a de nombreux problèmes. Du fait des occultations totales ou partielles, toutes les parties du corps ne sont pas visibles en même temps dans une image. Il n'est donc pas possible d'estimer l'ensemble des paramètres de pose à l'aide d'un unique point de vue. De plus, si le modèle prédit dans l'image qu'un membre est visible, il est possible que dans l'observation ce ne soit pas le cas. Il y a plusieurs raisons à cela :

- les contours ne sont pas extraits correctement,
- la visibilité prédite par le modèle concerne toujours l'image précédente et donc entre deux images consécutive le membre considéré peut avoir disparu.

Enfin, malgré le détecteur de contours optimisé que nous utilisons, il reste dans l'image des contours parasites qui contribuent à la distance de chanfrein et perturbent donc l'estimation des paramètres.

En pratique nous utilisons plusieurs caméras, ce qui permet d'augmenter le nombre de données. Chacun des points de vue permet de calculer une fonction de coût, qui lui est propre et qui contribue à la fonction de coût globale à minimiser. Si les caméras sont synchronisées et calibrées, la méthode décrite pour un point de vue donné peut s'appliquer à l'ensemble des points de vue. Les matrices Jacobiennes de chaque point de vue sont alors rassemblées dans une seule matrice pour n'obtenir qu'un unique Jacobien. La condition nécessaire est que les calculs doivent être effectués dans un repère commun ([99]). Nous augmentons ainsi le nombre de points pour l'estimation sans pour autant augmenter le nombre de paramètres à estimer.

Enfin, notons que les contours extrémaux vus d'un point de vue sont différents de ceux vus d'un autre. En réalité, ces contours sont la projection de points sur la surface du modèle, ces points étant différents selon le point de vue. Nous n'avons donc pas besoin de mettre en correspondance les contours observés dans une image avec ceux observés dans une autre.

6.3.2 Evaluation

Dans l'état de l'art, nous avons évoqué différents critères pour évaluer les algorithmes de suivi de mouvement.

Le degré d'automatisation : Si nous regardons l'ensemble des algorithmes que nous proposons, ils ne requièrent qu'assez peu d'intervention humaine. Lors de la cali-

bration, l'utilisateur doit vérifier que les marqueurs sont détectés dans les images. Lors de l'apprentissage du modèle du fond de l'image, il n'y a pas d'intervention humaine. L'estimation des dimensions de l'acteur nécessite une intervention lors de la création du squelette. Concernant l'initialisation et l'estimation du mouvement, l'intervention humaine se limite au chargement des données pour effectuer les traitements.

La vitesse d'exécution : Nous avons pris le parti d'avoir une méthode qui rend des résultats précis tout en ayant des temps de calcul acceptables. Notre implémentation actuelle pour l'estimation du mouvement permet de traiter une minute d'acquisition en deux heures. Les temps pourraient être réduits en optimisant le code et en parallélisant certaines étapes comme les traitements d'images (carte de contours, calcul de la distance de chanfrein, *etc.*).

Les contraintes d'acquisition : La méthode d'extraction de contours que nous avons proposé permet de s'affranchir (sous certaines conditions, comme la faible amplitude des mouvements) de l'utilisation des silhouettes. Nous avons donc la possibilité de travailler avec des environnements de capture peu contraints et où notamment la lumière peu varier. Dans les résultats que nous proposons, nous avons décidé d'habiller notre acteur avec une tenue homogène rouge. Cette tenue ne correspond pas à une contrainte pour notre système de capture du mouvement. Cette tenue, en plus d'être juste au corps, permet de différencier l'acteur du fond de l'image pour faciliter l'extraction des silhouettes. La couleur uniforme du costume n'a en réalité fait qu'augmenter la complexité de l'extraction des contours, un costume multicolore aurait facilité les traitements.

Le nombre de caméras Nous avons évoqué le coût du système. Nous avons vu que par caméra, nous devions compter environ 2000 €. Nous pourrions alors nous poser la question de la réduction du nombre de caméras afin de réduire le coût du système. Cependant, plus le nombre de caméras est réduit, plus le volume d'acquisition est faible ou la précision moindre. Lors de nos acquisitions à Rennes, nous avons constaté qu'en pratique quatre caméras étaient le minimum nécessaire pour faire des acquisitions utilisable pour de l'animation. Nous avons regardé le comportement du système sur un mouvement synthétique très simple lorsque le nombre de caméras diminuait. Nous donnons avec le graphique de la figure 6.22 les valeurs angulaires estimées avec deux et trois caméras. Sur des résultats synthétiques, nous pouvons observer une dégradation des estimations lorsque nous utilisons deux caméras bien que le suivi ne décroche pas complètement (c.f. figure 6.23). Sur des images réelles, nous avons testé la réduction du nombre de caméras sur une séquence acquise avec six caméras. Lorsque nous utilisons trois caméras les estimations sont mauvaises. Quatre caméras est donc le minimum requis pour effectuer l'estimation des paramètres sur une séquence réelle dans notre application.

Critères quantitatifs Les critères quantitatifs pour évaluer les résultats du suivi du mouvement sont difficiles à caractériser. Comment évaluer les résultats obtenus à l'aide

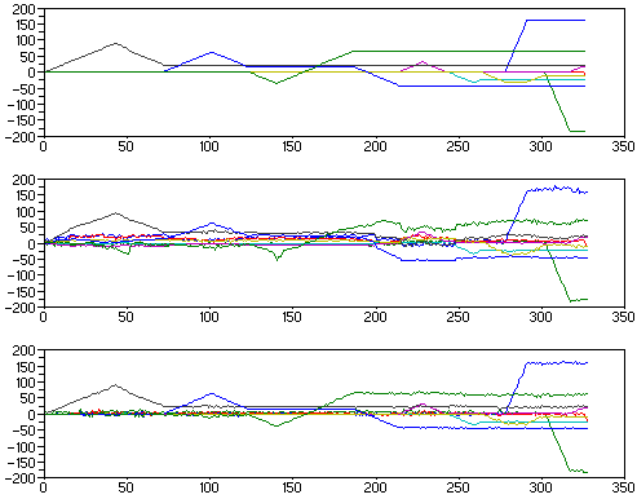


FIG. 6.22: Nous présentons ici les trajectoires angulaires (en degrés) de la vérité terrain (première ligne), celles estimées en utilisant deux caméras (seconde ligne) et enfin celles estimées avec trois caméras (dernière ligne). Les résultats de l'estimation utilisant deux caméras sont convaincants mais beaucoup plus bruités que ceux provenant de la configuration à trois caméras.

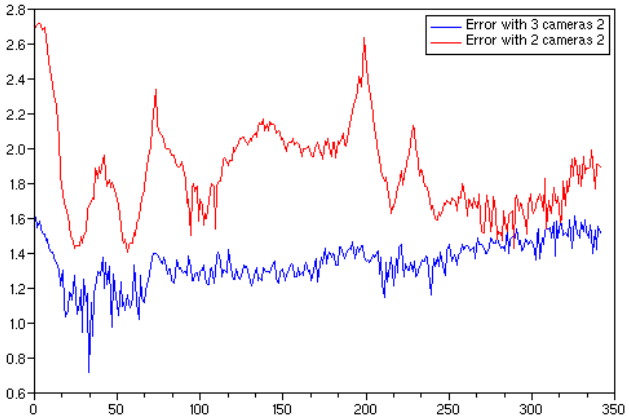


FIG. 6.23: Nous traçons l'erreur moyenne en pixels après estimation en utilisant deux caméras (en rouge, courbe du dessus) et trois caméras (courbe bleu, courbe du dessous). Les résultats sont plus stables en utilisant trois caméras.

de la capture du mouvement sans marqueurs par rapport à des résultats obtenus avec un système avec marqueurs. Plusieurs questions se posent :

- Comment synchroniser des données avec des fréquences d'échantillonnage différentes ?
- Quoi comparer ? Doit-on comparer les valeurs angulaires fournies par chaque système ? Doit-on comparer les positions des articulations au cours du temps ?

Peu de travaux discutent de cette évaluation du fait de sa complexité. Ce n'est que récemment que des universités ont commencé à proposer des bases de données permettant d'évaluer les résultats en les comparant à des données obtenues avec un système avec marqueurs ([62], [133]). Mais ce n'est que le début et c'est amené à se développer.

Nous avons évalué de manière quantitative les performances de nos algorithmes sur des séquences synthétiques. Nous avons comparé les valeurs angulaires estimées avec les valeurs de la vérité terrain. Nous avons aussi comparé les erreurs moyennes en pixel selon le nombre de caméras utilisées. Sur les séquences réelles, nous disposons maintenant de données VICON que nous pourrons utiliser pour effectuer la comparaison. Actuellement, nous avons donc évalué les résultats uniquement de manière qualitative (visuellement).

6.3.3 Deux extensions pour des améliorations

La méthode de minimisation que nous proposons dans le chapitre 5 ne prend en compte aucune contrainte lors de l'estimation des paramètres de pose. Dans ce cas rien n'empêche que plusieurs parties du corps se recouvrent en partie ou totalement. En pratique, cette situation arrive rarement. Il se peut cependant que les cônes associés aux bras du modèle se retrouvent à l'intérieur des cônes associés au tronc. Cette situation peut apparaître lorsque l'acteur, dans la séquence vidéo, passe ses bras le long du corps. Nous avons alors une ambiguïté lors de l'assignation des contours (lors de la phase de mise en correspondance) qui peut entraîner la situation décrite ci-dessus. Pour se prévenir de ce type de situation, nous avons mis en place une méthode de détection de collisions. Lorsqu'une collision entre deux membres est détectée, une pénalité dans la fonction de coût empêche l'un des membres de continuer à rentrer dans l'autre membre (sans pour autant bloquer l'évolution des paramètres). Nous présentons cette détection de collision au chapitre 7.

En outre, au cours de l'estimation du mouvement, nous avons des difficultés à suivre de manière efficace les pieds, les mains et la tête. En effet, le modèle de contours rectilignes et rigides est trop simpliste et modélise mal la réalité. Or, le suivi correct de ces membres donne une dimension plus réaliste au mouvement estimé. Nous proposons donc une extension permettant de suivre correctement l'ensemble de ces membres. Ce suivi spécifique permet de générer les trajectoires de chacun de ces membres au cours du temps. Cette génération de trajectoire permet un suivi plus robuste des membres mais permet aussi de contraindre le déplacement du modèle *3D* lors de l'estimation des paramètres. Nous présentons cette extension dans le chapitre 8.

Chapitre 7

Détection de collisions

Sommaire

Résumé

La méthode de suivi telle que présentée au chapitre 5 souffre d'un inconvénient majeur : rien n'empêche une partie du modèle *3D* de rentrer en collision avec une autre partie, de continuer sa trajectoire et donc de disparaître au cours du suivi. Afin de remédier à ce problème, nous proposons d'intégrer des contraintes de non-collision dans le suivi du mouvement. Nous adoptons une méthode de type événementielle traitant les collisions dès qu'elles apparaissent.

Pour effectuer la détection de collisions, nous utilisons des boîtes englobantes orientées permettant d'accélérer le calcul de collision. La détection de collision est une fonction binaire (collision ou non) qui, si elle était résolue telle quelle, rendrait inutilisable la minimisation mise en oeuvre pour effectuer le suivi. Pour passer d'une contrainte binaire à une contrainte douce, continue et dérivable, nous calculons une distance caractéristique du volume intersectant entre deux cônes. Nous explicitons la forme analytique de cette mesure. Cette dernière constitue alors une pénalité que nous utilisons dans la fonction de coût globale à minimiser pour estimer les paramètres de pose.

Enfin, nous proposons quelques perspectives à ce travail.

Introduction au chapitre

Le principe du suivi tel qu'abordé dans le chapitre 5 soulève plusieurs difficultés. Un problème majeur de la méthode est que rien ne peut empêcher une partie du modèle *3D* de venir « se cacher » dans une autre partie. Par exemple, si l'acteur pose sa main sur le torse, et si la détection de la main n'est pas correcte (c'est-à-dire que le nombre de vues où la main est détectée n'est pas suffisant), alors rien n'empêche la main de « rentrer » dans le torse. De manière générale ce type de difficulté est occasionné par une mauvaise association de contours. Prenons par exemple le cas où le bras de l'acteur passe le long du corps (c.f. figure 7.1). Dans les images, soit les contours associés au bras et au torse sont confondus au niveau du contact, soit les contours ne sont plus détectables (cas le plus courant, c.f. figure 7.1-a). Les contours du modèle au niveau du torse vont alors avoir tendance à s'associer avec les contours observés du bras. La mauvaise association des contours du torse entraîne alors une perte du suivi du bras (c.f. figure 7.1-b). Le bras du modèle va alors se retrouver caché par le torse et le calcul de visibilité détectera une absence du bras (c.f. figure 7.1-c). La pose du bras ne sera donc pas estimée. C'est pour éviter ces situations, que nous avons introduit des contraintes géométriques sur le modèle *3D*. Nous avons introduit une méthode de détection de collision évitant ainsi une interpénétration des parties du modèle *3D*, ce que nous traiterons dans ce chapitre.

Nous aborderons dans un premier temps un état de l'art ciblé sur la détection de collision ainsi que sur les méthodes que nous pouvons mettre en place pour réagir à la collision. Puis nous expliciterons le mécanisme que nous avons mis en place pour détecter les collisions entre les différentes parties de notre modèle *3D*. Nous montrerons ensuite comment rendre la contrainte de non-collision entre deux objets compatible avec la méthode de suivi que nous avons mise en place. Enfin, nous donnerons des perspectives à ce travail.

7.1 Etat de l'art

La détection de collision et le comportement à adopter pour éviter ou réagir à la collision sont des sujets qui sont abordés dans différents domaines. Que ce soit en robotique, en animation ou dans le cadre de l'interaction homme-machine, la détection de la collision permet de rendre une scène, une interaction, une trajectoire plus réaliste ou alors moins onéreuse (la collision en robotique est rarement appréciée !).

Dans le cadre du suivi de mouvement, la détection de collision permet d'éviter des situations telles que celles décrites dans l'introduction de ce chapitre.

La littérature sur les problèmes de détection de collision, que ce soit en robotique ou en animation est très dense. Beaucoup de travaux sont menés sur l'efficacité et l'aspect temps réel de la détection de collision ([47]). Pour la plupart des applications, la détection ou la prévention de collision doit être rapide (temps réel en robotique par exemple) et efficace. Dans notre cadre, la détection de collision se fait sur une scène

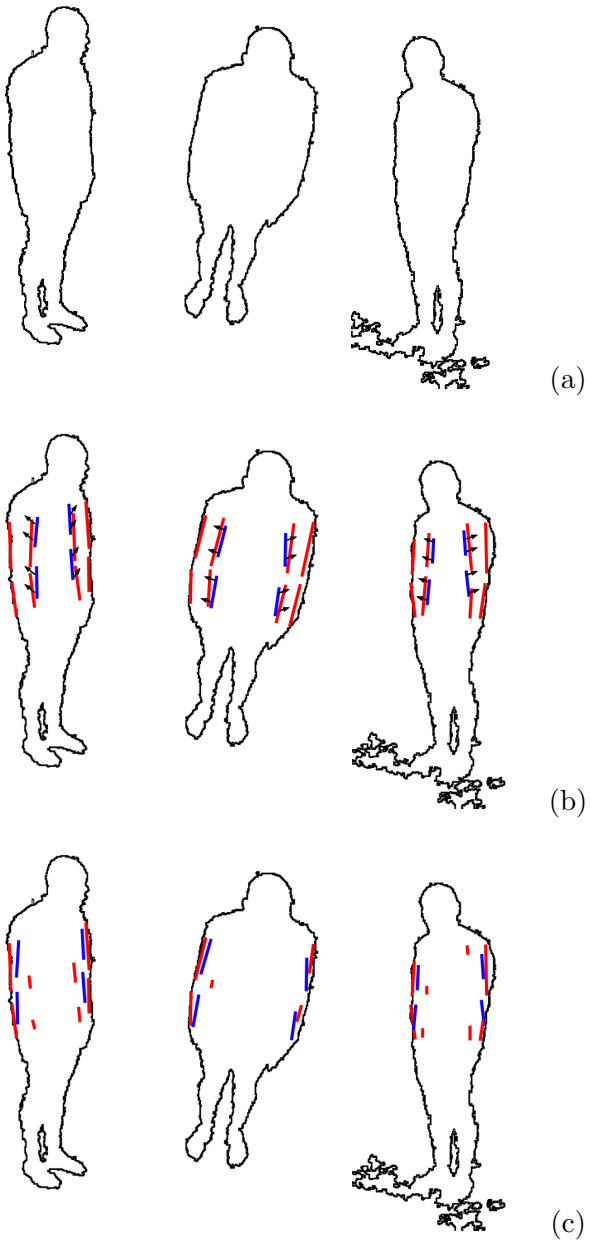


FIG. 7.1: Lorsque certains contours ne sont pas observables (a), l'absence de détection de collision peut amener à une mauvaise convergence de l'algorithme (b) et (c). Dans cet exemple, certains contours des bras ne sont pas correctement détectés et les contours du torse viennent se recaler sur de mauvais contours.

simple avec peu de primitives géométriques. Nous aborderons donc dans cet état de l'art les techniques les plus adaptées à notre approche. Plus précisément, nous aborderons les travaux menés en animation ou en rendu de scènes pour des objets simples. En robotique, le problème de la détection de collision se ramène au problème de planification de trajectoires sans collisions. Nous n'aborderons pas cet aspect là dans le cadre de cet exposé. Le lecteur pourra se référer à [92] pour un aperçu sur les techniques de planification de mouvements sans collision.

Lorsque plusieurs objets bougent dans une scène, il est fortement probable que ceux-ci s'interpénètrent à un moment ou à un autre au cours du temps. Ce n'est généralement pas une situation voulue, tout du moins pour la modélisation ou le rendu de scènes réalistes. Il y a deux problèmes ici qui entrent en jeu : détecter la collision entre deux objets et déterminer la réponse à cette collision. Beaucoup d'approches ont tenté de résoudre les deux problèmes. Le premier est purement d'ordre cinématique. Il prend en compte la position des objets dans la scène et calcule une relation d'ordre entre ceux-ci. Si deux objets sont plus proches qu'un seuil donné alors ils sont en collision. Le deuxième problème est plus d'ordre de la dynamique. Lorsque deux objets sont en collision, comment faut-il les faire réagir pour qu'ils s'éloignent l'un de l'autre. Ce deuxième problème est régi par des lois physiques (choc élastique ou mou par exemple). Nous présenterons dans une première partie les méthodes couramment utilisées en animation pour la détection de collision. Nous aborderons dans une deuxième partie les réponses possibles à la collision, adaptées à notre approche.

7.1.1 La détection de collision

De manière générale, la détection de collision dans une scène graphique consiste à déterminer s'il existe une intersection entre les primitives géométriques composant la scène. Et généralement ces primitives sont des triangles.

Quelque soit la méthode de détection mise en place, la complexité de la détection est quadratique (que ce soit par rapport au nombre de triangles par objet ou au nombre d'objets dans la scène). D'autre part, la détection de collision entre tous les triangles de la scène est une méthode brute de force et inefficace pour du calcul rapide ou temps réel de collision. Deux simplifications sont donc utilisées pour accélérer les calculs : la division de la scène en sous-espaces pour n'effectuer les calculs que sur des éléments proches les uns des autres et l'utilisation de primitives simples englobantes pour calculer simplement les intersections. Ces boîtes englobantes peuvent être de type sphérique ou parallélépipèdique. La scène est alors décomposée en un ensemble de boîtes organisé et hiérarchisé. Beaucoup de travaux ont été menés pour rendre efficace ce partitionnement de la scène. Ces travaux portent surtout sur les structures utilisées pour la hiérarchisation des boîtes. Ces structures peuvent être de type *cone trees*, *k-d trees*, *octrees*, etc. D'autres méthodes comme celles de type BSP (*binary space partitionning* [112]) existent. Nous n'entrerons pas dans les détails de cette organisation car nous n'en n'utiliserons pas (notre scène est trop peu complexe). Précisons juste que ces méthodes hiérarchiques sont très efficaces pour des tests de réjection, c'est-à-dire pour détecter

l'absence de collision entre deux objets donnés. Cependant lorsque deux objets sont en contact, et pour déterminer l'ensemble des points de contact, ces méthodes nécessitent d'affiner la hiérarchie des boîtes englobantes et donc d'augmenter de manière significative les temps de calcul ([66]).

Dans le paragraphe précédent, nous avons abordé le problème de l'organisation de la scène pour effectuer le calcul de collision. La détection de collision en elle-même peut se faire de manière statique ou dynamique. Dans le premier cas, il s'agit d'effectuer une détection de la collision effective entre deux objets. Cette détection statique est la plus simple à mettre en place. Il s'agit de calculer les intersections entre tous les triangles décrivant la scène ([106], [64]). Dans le second cas, il s'agit d'effectuer une détection a priori de la collision. Il s'agit alors de prédire la collision ou non entre deux objets à très court terme. Cette seconde classe de méthodes est beaucoup plus efficace mais plus complexe à mettre en place. En effet, nous devons connaître à tout instant un modèle de déplacement de nos objets dans la scène. [122] propose une méthode permettant de détecter de manière efficace et continue la collision entre plusieurs objets sans connaître le modèle de mouvement précis de l'objet mais en interpolant celui-ci entre deux instants donnés (avec un mouvement de type vissage entre deux positions connues).

Nous pouvons noter que d'autres techniques de type champ de distance ([51]), stochastiques ([65]) ou encore des méthodes exploitant les GPUs (*Graphical Processing Units*) des cartes graphiques ([47]) existent et sont en pleine expansion.

Enfin, pour une vue d'ensemble des méthodes utilisées les plus couramment, le lecteur pourra se référer à [78] ou encore [93].

Pour des raisons de simplicité et de rapidité, nous nous sommes limités à l'utilisation d'une méthode de détection de collision statique. Afin de rendre les calculs de collision efficaces, nous utiliserons des boîtes englobantes orientées (dénotées dans la suite de l'exposé par OBB pour « oriented bounding box » [59]). Il s'agit de boîtes englobantes, de forme parallélépipèdique, munies d'un repère dont l'origine est située au centre de la boîte. Celles-ci sont intensivement utilisées en rendu de scène ([8]). Le fait d'orienter les boîtes englobantes permet d'effectuer de manière efficace le calcul de collision comme nous allons le voir dans la suite de ce chapitre.

7.1.2 La réponse à la collision

Une fois la collision détectée, il faut calculer la réaction des objets. Ceux-ci peuvent continuer de s'interpénétrer, rebondir l'un contre l'autre, adhérer l'un à l'autre. Nous allons ici aborder les types de réponses utilisés en rendu haptique pour l'interaction homme-environnement virtuel. En effet, l'aspect retour de force des systèmes haptiques se rapproche beaucoup du cadre dans lequel nous effectuons la détection de collision. Dans les deux cas, nous voulons faire interagir deux objets en admettant une interpénétration plus ou moins grande et un retour de force plus ou moins intense. Cependant, contrairement à l'interaction haptique, nous ne considérons pas de lois de frottement entre les objets, ni de modélisation précise de l'interaction entre les objets.

Nous pouvons voir deux points de vue pour la réponse à la détection de collision :
- La collision a lieu et le système réagit en conséquence. Il s'agit alors d'imposer une contrainte de déplacement aux objets pour qu'ils ne soient plus en collision.
- La collision est empêchée a priori en contraignant le déplacement des objets dans la scène.

Le premier point de vue est celui qui est utilisé en rendu haptique, où le système doit rendre compte à l'utilisateur d'une interaction et non empêcher ce dernier d'effectuer cette interaction.

C'est le deuxième point de vue qui est le plus largement usité pour le suivi de mouvement. En effet, la plupart des approches intégrant des contraintes sur le déplacement utilisent un apprentissage sur les contraintes articulaires ou utilisent des contraintes articulaires bio-mécaniques pour forcer le mouvement à rester dans des limites bio-mécaniques acceptables ([70]). Cependant ces approches ne suffisent pas pour éviter les interpénétrations d'objets et donc des échecs probables du suivi de mouvement. En effet, les contraintes bio-mécaniques sont généralement des valeurs moyennes pour les limites articulaires. Elles ne prennent pas en compte les différences de morphologie humaine. L'apprentissage des contraintes doit donc être personnalisé.

D'autre part, beaucoup de travaux évitent ce problème en effectuant un apprentissage des poses possibles et donc vérifient si la pose estimée coïncide avec une pose apprise ([2]). Ces dernières méthodes limitent cependant le nombre de mouvements qu'il est possible de suivre, puisque la base d'apprentissage doit correspondre au mouvement suivi.

Pour éviter les phases d'apprentissage (que ce soit des limites articulaires ou des poses possibles) ou la limitation des mouvements, nous avons donc décidé de mettre en place une méthode de réaction à la collision et non de prévention de celle-ci.

7.2 Méthode

Comme nous avons pu le voir (chapitre 5), le suivi du mouvement s'effectue en optimisant les paramètres de pose d'un modèle *3D*. Cette optimisation s'effectue par minimisation itérative de l'erreur entre les contours images et les contours projetés du modèle *3D*. La prise en compte correcte des contraintes de non collision oblige à évaluer les collisions sur l'ensemble des parties du corps et à chaque itération. Or effectuer le calcul de la distance entre chaque partie du corps à tout instant peu s'avérer coûteux. Afin de rendre les calculs plus rapides, la résolution des contraintes de non pénétration s'effectue en deux étapes :

La détection de collision est effectuée à chaque itération sur l'ensemble des cônes. Afin de déterminer la collision ou non entre deux objets, nous avons décidé d'adapter et d'utiliser les OBBs. La simplicité de notre scène ainsi que des primitives géométriques de notre modèle nous permet d'éviter l'utilisation d'une structure organisée de boîtes. Chaque cône du modèle *3D* est englobé dans un parallélépipède orienté, dont les dimensions sont données par celles du cône (c.f. figure 7.2). La

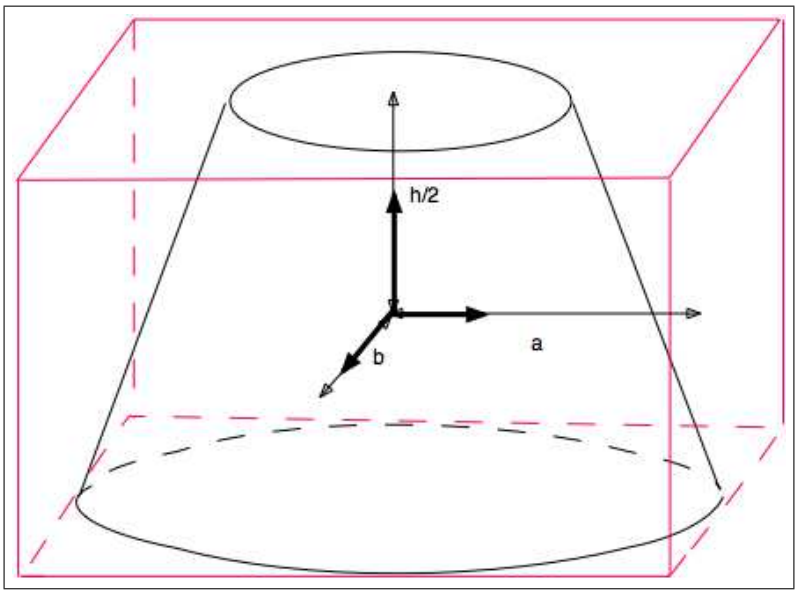


FIG. 7.2: Chacune des parties du modèle est munie d'une boîte englobante orientée. Les dimensions de celle-ci sont données par la base du cône elliptique et sa demi-hauteur.

collision est alors calculée sur ces OBBs. Comme nous le verrons dans le paragraphe 7.3, il s'agit d'une opération peu coûteuse et qui peut donc être effectuée aussi souvent que nécessaire.

Le calcul de la distance d'interpénetration n'est effectué que si le test de collision est positif. Il s'agit de calculer une distance caractérisant le volume intersectant de deux cônes. La distance ainsi calculée est utilisée dans la fonction de coût du suivi de mouvement comme une pénalité sur l'estimation de la position des cônes concernés. Cette pénalité (décrite plus bas) diminue lorsque les cônes s'éloignent les uns des autres et est nulle lorsqu'il n'y a plus de contact entre eux. Nous verrons que cette opération est coûteuse. Il est donc nécessaire de réduire le nombre de fois où le calcul est effectué.

Dans la suite de ce chapitre, nous allons expliciter la méthode mise en place pour détecter les collisions. Puis nous définirons et établirons la distance d'interpénétration entre deux cônes. Nous décrirons la fonction de pénalité utilisée pour prendre en compte la contrainte de non pénétration des différentes parties du corps. Dans une troisième partie, nous établirons la Jacobienne de la fonction de pénalité. Enfin, nous présenterons quelques résultats et discuterons des extensions possibles de la méthode.

7.3 La détection de collision

De manière générale, la détection de collision consiste à calculer une distance entre deux objets et à déterminer si cette distance est plus petite qu'une distance critique. Cette distance est calculée entre des points, des arrêtes ou des polyèdres. Dans le cas

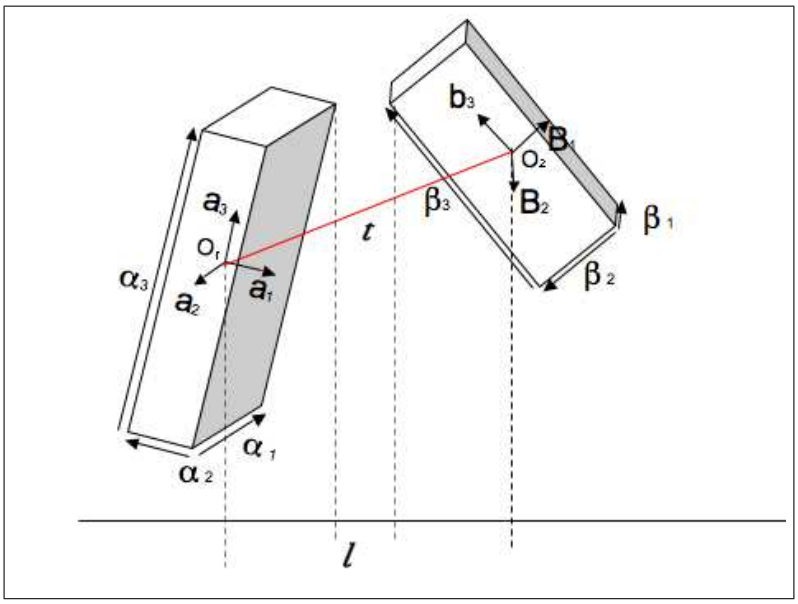


FIG. 7.3: Deux boîtes englobantes ne sont pas en contact s'il existe un axe sur lequel les projections axiales des deux boites sont disjointes.

des OBBs, il s'agit de calculer la distance entre chaque arrête et chaque face des deux volumes. Mais il s'agit d'un calcul coûteux et complexe. L'approche que nous proposons s'appuie sur les travaux décrits dans [59]. Pour détecter une collision entre deux OBBs, les auteurs s'appuient sur la notion d'axe séparateur (que nous allons expliciter ci-dessous). Ils établissent un théorème simplifiant la détection de la collision. Dans un premier temps nous définirons la notion d'axe séparateur puis nous verrons son application au cas du squelette articulé.

L'axe séparateur Un axe L orienté par un vecteur $\boldsymbol{l}$ est dit séparateur pour deux boîtes englobantes si leurs projections sur cet axe sont disjointes. Si deux boites englobantes (A_1 et A_2) sont disjointes alors il existe un axe séparateur orthogonal à :

- une face de A_1,
- une face de A_2,
- une arrête de chacune des boîtes.

$\boldsymbol{l}$ doit donc vérifier la contrainte suivante :

$$|\boldsymbol{t}.\boldsymbol{l}| > 1/2(\sum_{i=1}^{3} \alpha_i \boldsymbol{a}_i \cdot \boldsymbol{l} + \sum_{i=1}^{3} \beta_i \boldsymbol{b}_i \cdot \boldsymbol{l}), \tag{7.1}$$

où α_i et β_i (pour $i \in [0..2]$) sont les dimensions des boîtes englobantes ; $\boldsymbol{a}_i$ et $\boldsymbol{b}_i$ sont les vecteurs unitaires des repères associés à chacune des boites. $\boldsymbol{t}$ est le vecteur directeur de l'axe reliant les centres des repères de chacune des boites (O_1 et O_2). Si on trouve un axe $\boldsymbol{l}$ vérifiant l'inégalité alors les boîtes ne sont pas en collision (c.f. figure 7.3).

De manière intuitive, l'équation (7.1) permet de vérifier que la distance entre les centres des boîtes projetés sur L est plus grande que la somme des « rayons » de chaque boîte projetée sur ce même axe.

Théorème de l'axe séparateur La recherche d'un axe L vérifiant l'inégalité précédente peut s'avérer extrêmement complexe. Cependant, [59] propose de résoudre le problème proposant et démontrant le théorème suivant :

Théorème 1 *Si deux boîtes englobantes orientées sont disjointes alors il existe un axe séparateur* $\boldsymbol{l} = \boldsymbol{w} \times \boldsymbol{u}$ *où* $\boldsymbol{w}$ *et* $\boldsymbol{u}$ *sont deux vecteurs pris parmi les axes des repères des deux boîtes.*

Le choix de $\boldsymbol{w}$ et $\boldsymbol{u}$ nous amène donc à choisir parmi 15 possibilités pour l'axe $\boldsymbol{l}$:
- $\boldsymbol{a}_i \times \boldsymbol{a}_j$ pour $i \neq j$,
- $\boldsymbol{b}_i \times \boldsymbol{b}_j$ pour $i \neq j$,
- $\boldsymbol{a}_i \times \boldsymbol{b}_j$.

Il s'agit alors simplement de tester l'inégalité de l'équation (7.1) pour l'ensemble de ces 15 axes. Dès qu'un des axes vérifie l'inégalité, il n'y a pas de collision. Pour la preuve du théorème, le lecteur pourra se référer aux pages 10 et 11 de [59].

Le choix de ces axes particuliers permet de simplifier l'équation (7.1). Par exemple, si nous prenons $\boldsymbol{l} = a_2 \times a_3$, l'inégalité devient :

$$|\boldsymbol{t}.\boldsymbol{l}| > 1/2(\alpha_1 \boldsymbol{a}_1 \cdot \boldsymbol{a}_1 + \boldsymbol{a}_1 \cdot \sum \beta_i \boldsymbol{b}_i). \tag{7.2}$$

En testant les cas triviaux en premier, le test de réjection de collision s'avère très peu coûteux. En effet, environ 200 opérations sont nécessaires pour tester l'ensemble des 15 axes. En général, il suffit de tester l'un des axes pour déterminer s'il n'y a pas de collision, donc les 200 opérations ne sont effectuées que très rarement.

Application au squelette articulé Lors du suivi du mouvement, il s'agit de tester à chaque instant si deux parties du corps sont en collision. Chacun des cônes elliptiques modélisant chacune des parties du corps est englobé dans une OBB. Si nous reprenons les notations de la figure (7.2), la boite englobante aura comme dimension ($2a$, $2b$, h).

A chaque itération, le test de collision est effectué sur l'ensemble des parties du corps. Le nombre de tests est alors de l'ordre de N^2, où N est le nombre de cônes modélisant le corps, ce qui est relativement faible et donc ne prends que très peu de temps. Nous adoptons une résolution des collisions de type événementielle ce qui permet de résoudre les collisions de manière continue. Plus précisément, à chaque modification du squelette nous faisons un test de collision. Cela permet de rester dans des contraintes linéaires pour les résoudre.

Cependant, nous devons prendre des précautions pour effectuer le test de collision dans le cadre d'une chaîne articulée. En effet, le test de collision s'avère positif entre

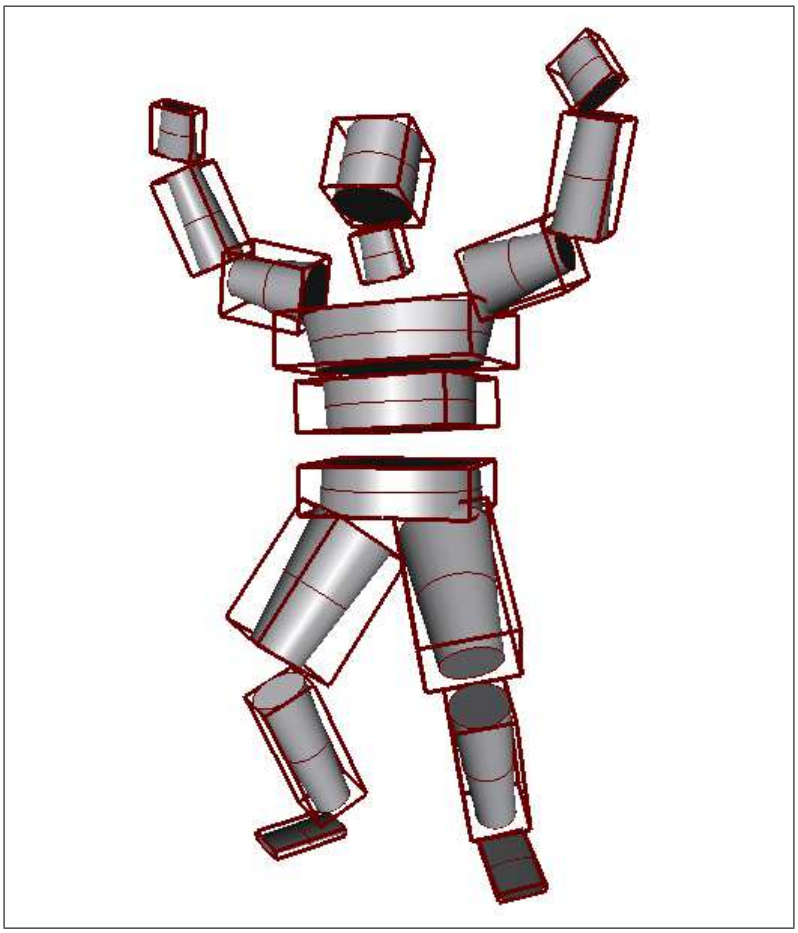


FIG. 7.4: Chacune des parties du modèle *3D* est munie d'une boîte englobante.

deux cônes liés par une articulation et ce quelque soit la position relative des cônes (sauf s'ils sont alignés). Par exemple, dès lors que l'avant bras est plié, les OBBs ne sont plus disjointes au niveau de l'articulation. Il y a donc collision et donc un terme de pénalité dans la minimisation empêchant l'avant bras de se plier complètement. Nous verrons dans la suite de ce chapitre comment nous contournons cette difficulté.

7.4 Le calcul de la distance d'interpénétration

Le calcul précédent permet de conclure quant à la possible collision entre deux cônes. En effet, le calcul de collision est effectué sur les boîtes englobantes qui représentent au mieux les cônes mais pas exactement. Il se peut donc qu'une détection de collision soit une fausse alarme. Cette erreur est détectée lors du calcul de la distance d'interpénétration des cônes.

La distance d'interpénétration est une quantité caractérisant le volume intersectant

deux cônes. Nous en verrons une définition analytique plus bas. Ce calcul de distance permet d'introduire une pénalité dans la fonction d'énergie à minimiser pour effectuer le suivi. Cette pénalité est déterminée de sorte que la contrainte de non-collision, qui est binaire, devienne une contrainte continue. Plus précisément, la contrainte est nulle lorsqu'il n'y a pas de collision et augmente de manière continue dès que deux cônes sont en contact.

Le calcul de la distance d'interpénétration prend en compte la distance entre les axes principaux des cônes mais aussi les rayons de ces cônes. Nous pouvons définir la distance d'interpénétration de la manière suivante :

$$D_{12} = -d_{axes} + R_1 + R_2, \tag{7.3}$$

où d_{axes} est la distance entre les axes principaux, R_1 et R_2 sont les rayons des deux cônes. D'autre part, nous considérons cette distance comme nulle si $d_{axes} > R_1 + R_2$ (il n'y a pas de collisions).

Dans ce paragraphe, nous expliciterons les calculs de d_{axes}, R_1 et R_2. Cependant, nous avons noté plus haut que le cas de deux cônes liés par une articulation ne pouvait être considéré comme celui de deux cônes disjoints. Nous traiterons donc séparément les deux cas.

7.4.1 Calcul de la distance entre deux cônes

Nous allons expliciter ici le calcul de la distance entre les axes principaux de deux cônes. Il s'agit en fait de calculer la distance entre deux segments dans l'espace. Si les deux cônes sont liés l'un à l'autre par une articulation, la distance est tout le temps nulle. Nous adaptons donc les calculs pour ce cas particulier.

Deux cônes disjoints dans la chaîne articulaire Soit deux cônes d'axes principaux $[AB]$ et $[CD]$. Chacun des points (A, B, C et D) est situé à l'intersection entre l'axe et la base (ou le sommet du cône) (c.f. figure 7.8). Pour effectuer le calcul de la distance entre les deux cônes, nous allons effectuer le calcul de la distance entre les deux segments ($[AB]$ et $[CD]$). Les coordonnées de chacun des points doivent donc être exprimées dans un repère commun que nous choisirons comme étant celui du monde.

Soit $P_1 \in [AB]$ et $P_2 \in [CD]$ deux points. La distance entre les segments $[AB]$ et $[CD]$ est telle que $\|\overrightarrow{P_1P_2}\|^2$ soit la plus petite possible. Pour calculer la distance, il s'agit de minimiser $\|\overrightarrow{P_1P_2}\|^2$ en fonction de la position de P_1 et P_2. Mais nous pouvons aussi effectuer le calcul de manière explicite, ce que nous allons développer maintenant.

Quatre cas existent pour le calcul de la distance :
- La distance la plus courte entre les deux segments est telle que P_1 **et** P_2 soient confondus avec les extrémités des segments (figure 7.5 (a)),
- P_1 **ou** P_2 est confondu avec l'une des extrémités des segments (figure 7.5 (b)),

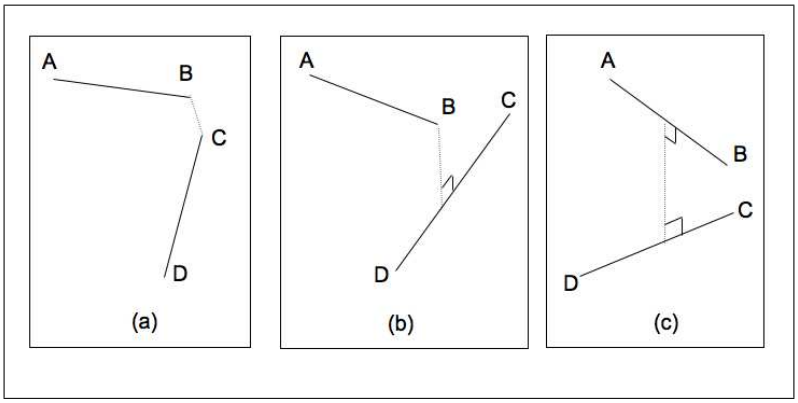


FIG. 7.5: Plusieurs cas existent pour calculer la distance entre deux segments.

– Les deux points les plus proches sont situés « à l'intérieur » des segments (figure 7.5 (c)),
– Les segments sont quasiment parallèles.

Nous allons maintenant résoudre analytiquement ces cas.

Le cas du parallélisme : il revient en fait au cas n°2 en fixant, par exemple, $P_1 = A$.

Précision : La vérification du parallélisme des segments s'effectue de la manière suivante : soit $\boldsymbol{u}$ et $\boldsymbol{v}$ les vecteurs directeurs des deux segments, dont les coordonnées sont exprimées dans un repère commun (celui du monde). Si $\|\boldsymbol{u}\|^2\|\boldsymbol{v}\|^2 = \|\boldsymbol{u}\cdot\boldsymbol{v}\|\|\boldsymbol{v}\cdot\boldsymbol{u}\|$ alors $[AB]$ et $[CD]$ sont parallèles.

Le cas général : La distance entre les segments $[AB]$ et $[CD]$ est telle que le vecteur $\overrightarrow{P_1P_2}$ est orthogonal aux deux segments. Il s'agit alors de trouver P_1 et P_2 tels que

$$\left\{\begin{array}{rcl} \boldsymbol{u}\cdot\overrightarrow{P_1P_2} & = & 0 \\ \boldsymbol{v}\cdot\overrightarrow{P_1P_2} & = & 0. \end{array}\right. \tag{7.4}$$

Si nous posons s_c et t_c tels que $\overrightarrow{AP_1} = s_c\boldsymbol{u}$ et $\overrightarrow{CP_2} = t_c\boldsymbol{v}$ et $\boldsymbol{w}_0 = \overrightarrow{AC}$, alors $\overrightarrow{P_1P_2} = -s_c\boldsymbol{u} + \boldsymbol{w}_0 + t_c\boldsymbol{v}$. En substituant dans l'équation (7.4), s_c et t_c doivent vérifier les conditions suivantes :

$$\left\{\begin{array}{rcl} 0 & = & -s_c\|\boldsymbol{u}\|^2 + \boldsymbol{w}_0\cdot\boldsymbol{u} + t_c\boldsymbol{u}\cdot\boldsymbol{v} \\ 0 & = & -s_c\boldsymbol{v}\cdot\boldsymbol{u} + \boldsymbol{w}_0\cdot\boldsymbol{v} + t_c\|\boldsymbol{v}\|^2 \end{array}\right. \tag{7.5}$$

La résolution du système d'équations précédent nous donne les deux coefficients s_c et t_c.

$$\left\{\begin{array}{rcl} t_c & = & \frac{(\boldsymbol{w}_0\cdot\boldsymbol{u})(\boldsymbol{u}\cdot\boldsymbol{v}) - (\boldsymbol{w}_0\cdot\boldsymbol{v})\|\boldsymbol{u}\|^2}{(\boldsymbol{u}\cdot\boldsymbol{v})(\boldsymbol{u}\cdot\boldsymbol{v}) - \|\boldsymbol{u}\|^2\|\boldsymbol{v}\|^2} \\ \\ s_c & = & \frac{(\boldsymbol{w}_0\cdot\boldsymbol{v})(\boldsymbol{u}\cdot\boldsymbol{v}) - (\boldsymbol{w}_0\cdot\boldsymbol{u})\|\boldsymbol{v}\|^2}{(\boldsymbol{u}\cdot\boldsymbol{v})(\boldsymbol{u}\cdot\boldsymbol{v}) - \|\boldsymbol{u}\|^2\|\boldsymbol{v}\|^2} \end{array}\right. \tag{7.6}$$

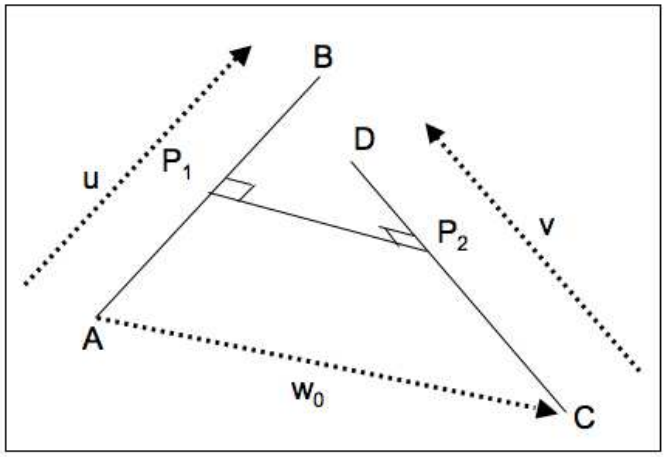


FIG. 7.6: La distance $\|P_1P_2\|$ entre deux segments est telle que le vecteur $\overrightarrow{P_1P_2}$ est orthogonal à $\boldsymbol{u}$ et $\boldsymbol{u}$.

s_c et t_c doivent être compris dans l'intervalle [0..1]. Le cas contraire, c'est que P_1 et/ou P_2 sont à l'extérieur des segments considérés. Le cas échéant, les points à l'extérieur sont fixés à l'extrémité (du segment) la plus proche (nous nous ramenons alors au premier ou au second cas). Il suffit alors de calculer la distance entre l'extrémité du segment et le second segment, ce qui revient à poser $s_c = 0$ ou 1 et/ou $t_c = 0$ ou 1.

Deux cônes liés par une articulation Dans ce cas, la distance entre les axes principaux telle que définie dans le paragraphe précédent est toujours nulle. Il a donc fallut trouver un nouveau critère pour évaluer la distance entre deux cônes. Il nous a semblé raisonnable d'estimer la distance entre le milieu de l'axe principal d'un des cônes et le second cône (figure 7.8 (b)). Ce choix est fait pour deux raisons :

- le calcul d'interpénétration sera simplifié (comme nous le verrons ci-dessous)
- si nous prenons le cas de la jambe ou du bras (qui sont les 2 membres concernés dans la majorité des cas), on peut remarquer que la flexion entraîne un écrasement des muscles au niveau de l'articulation, non explicitement modélisé par notre modèle. Pour modéliser plus correctement la flexion, nous devons permettre aux cônes de rentrer en collision dans une certaine mesure. Cependant, les deux cônes ne doivent pas pouvoir se retrouver dans la situation illustrée par la figure 7.7 où un des cônes est caché par l'autre.

Pour calculer cette distance, il suffit de calculer la distance entre un point et un segment et donc de se ramener aux cas étudiés plus haut.

7.4.2 Calcul de la distance d'interpénétration

Dans le paragraphe précédent, nous avons vu comment calculer la distance entre les axes principaux des cônes. Cependant, cela ne suffit par pour caractériser le vo-

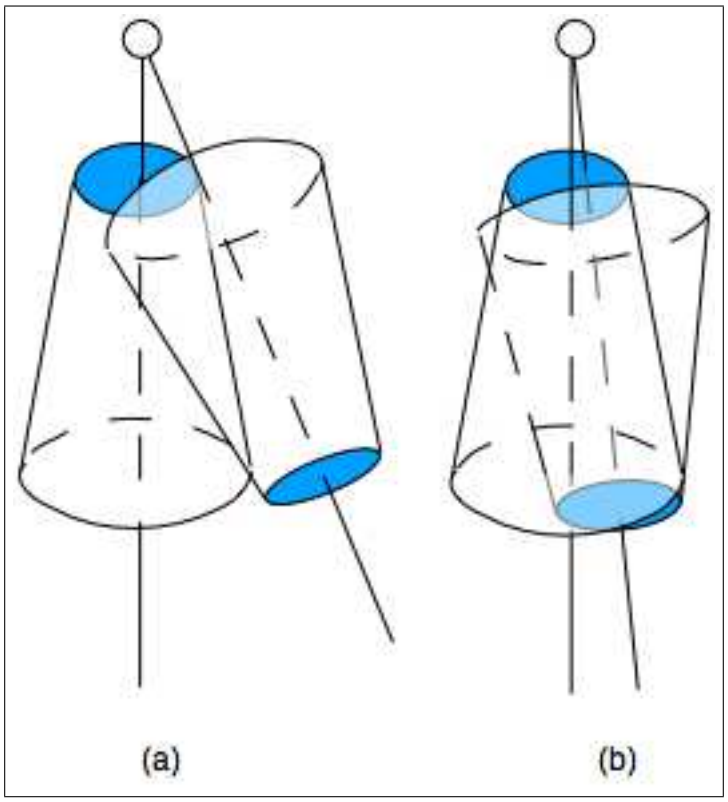


FIG. 7.7: Lorsque deux cônes sont liés par une articulation, l'interpénétration apparaît très vite (a). La contrainte de collision doit donc être relâchée de sorte à n'empêcher que le cas où les deux cônes seraient totalement l'un dans l'autre (b).

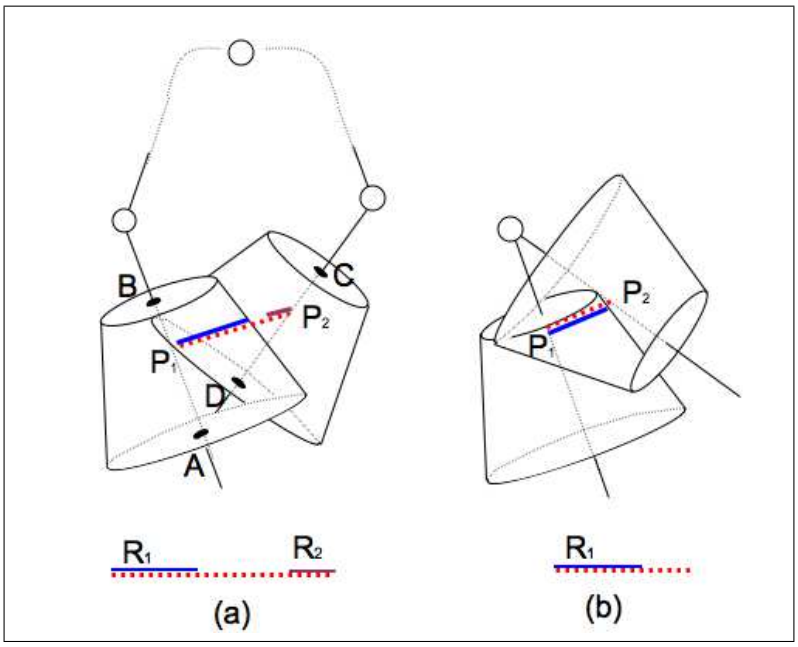


FIG. 7.8: La distance d'interpénétration est calculée en fonction de la distance entre les axes principaux des cônes et les rayons des cônes projetés sur la droite (P_1P_2).

lume intersectant des deux cônes. Pour rendre compte de l'intersection des cônes, nous définissons analytiquement la distance d'interpénétration de la manière suivante :

$$D_{12} = \|\overrightarrow{P_1P_2}\| - R_1 - R_2, \tag{7.7}$$

où R_1 et R_2 sont les rayons des cônes calculés en P_1 (respectivement en P_2) (c.f. figure 7.6).

Nous avons explicité $\|\overrightarrow{P_1P_2}\|$ dans le paragraphe précédent, nous allons maintenant calculer R_1 et R_2.

Nous avons vu dans le chapitre 4 que le cône peut être paramétré de la manière suivante :

$$\boldsymbol{X}(\theta, z) = \begin{pmatrix} a(1+kz)\cos(\theta) \\ b(1+kz)\sin(\theta) \\ z \end{pmatrix}. \tag{7.8}$$

Pour une hauteur z donnée, et une orientation θ donnée, le rayon a pour expression :

$$R_i^2 = (a^2(1+kz_i)^2\cos(\theta_i)^2 + b^2(1+kz_i)\sin(\theta_i)^2, \tag{7.9}$$

avec i $(i \in 1, 2)$, $z_1 = s_c\|\overrightarrow{AB}\|$ et $z_2 = t_c\|\overrightarrow{CD}\|$. La seule inconnue à déterminer est donc θ_i. De la même manière que précédemment, nous distinguons le cas de deux cônes disjoints et de deux cônes liés par une articulation.

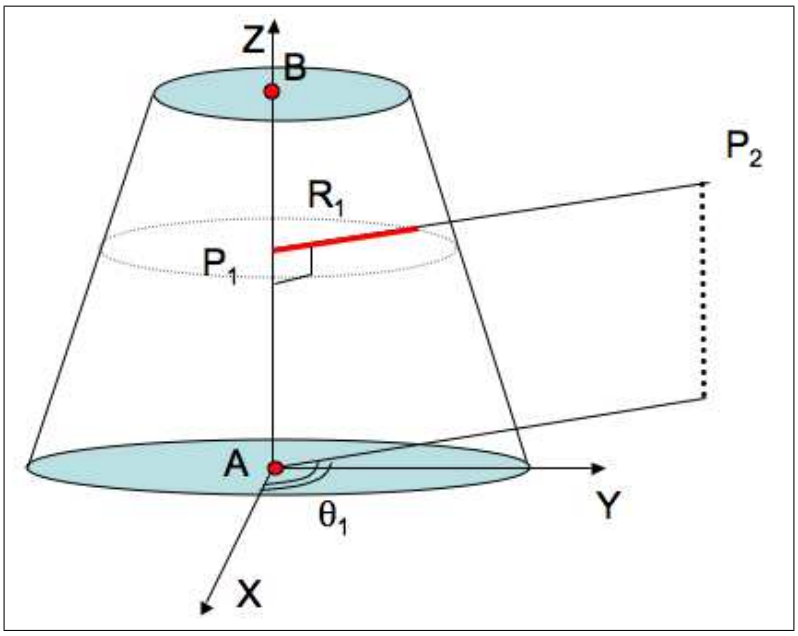


FIG. 7.9: Le calcul de R_1 nécessite de calculer θ_1.

Deux cônes disjoints dans la chaîne articulaire Nous allons nous servir des résultats précédents pour déterminer θ_i qui est l'angle formé par le vecteur du repère $\boldsymbol{x}$ et le vecteur $\overrightarrow{P_1P_2}$. Nous avons déterminé les coordonnées de P_1 et P_2 dans un repère commun (celui du monde). Pour calculer θ_i, nous devons effectuer un changement de repère pour que P_1 et P_2 soient exprimés dans le repère du cône $i \in [1, 2]$. De manière équivalente, nous pouvons exprimer le vecteur $\overrightarrow{P_1P_2}$ dans le repère du cône $i \in [1, 2]$.

Si $\mathbf{R}_i$ est la matrice d'orientation du cône i par rapport au référentiel de référence, alors nous avons :

$$\boldsymbol{t}_i = \mathbf{R}_i^{-1}\overrightarrow{P_1P_2},$$

où $\boldsymbol{t}_i$ est le vecteur des coordonnées du vecteur $\overrightarrow{P_1P_2}$ dans le repère du cône i. Nous pouvons ainsi calculer $\cos(\theta_i)$ et $\sin(\theta_i)$:

$$\begin{aligned} \cos(\theta_i) &= \boldsymbol{t}_i.x/\|\overrightarrow{P_1P_2}\|, \\ \sin(\theta_i) &= \boldsymbol{t}_i.y/\|\overrightarrow{P_1P_2}\|. \end{aligned} \tag{7.10}$$

Nous avons maintenant tous les éléments pour calculer l'interpénétration entre les deux cônes. Si $D_{12} = \|\overrightarrow{P_1P_2}\| - R_1 - R_2 > 0$ alors il n'y a pas d'interpénétration entre les cônes.

Remarque : Si P_1 et/ou P_2 sont confondus avec les extrémités des segments, R_1 et/ou R_2 sont considérés comme nuls.

Deux cônes liés par une articulation Dans ce cas, le calcul s'avère plus compliqué si nous voulons le faire de manière exacte. En effet, comme nous pouvons le voir sur la figure 7.6, R_2 est difficilement calculable. Nous calculons donc R_1 (avec le même calcul que celui explicité plus haut) et nous définissons la distance d'interpénétration de la manière suivante : $D_{12} = \|\overrightarrow{P_1P_2}\| - R_1$. Il s'agit de faire en sorte que le milieu de l'axe principale du cône 2 ne puisse pas pénétrer à l'intérieur du cône 1 (c.f. figure 7.7).

7.5 Calcul de la contrainte et de sa Jacobienne

Dans les paragraphes précédents, nous avons explicité la fonction d'interpénétration entre deux cônes. Nous allons maintenant montrer comment nous intégrons cette distance comme critère de contrainte dans l'algorithme de suivi. Nous présenterons dans un premier temps différentes méthodes que nous pourrions employer pour contraindre le déplacement des différentes parties du modèle *3D*. Puis nous donnerons et justifierons notre choix de contrainte. Enfin, nous établirons la Jacobienne de la contrainte mise en place, nécessaire à son intégration dans la méthode de minimisation pour effectuer le suivi.

7.5.1 Les fonctions de répulsion

Nous avons vu que l'animation graphique ou le rendu haptique nécessitait de prendre en compte les collisions. Ce sont des domaines où les recherches sur ce problème sont très actives. En suivi du mouvement, le problème est souvent considéré comme acquis. Peu de travaux portent sur ce problème. Nous allons présenter différentes méthodes issues essentiellement de la littérature sur le rendu haptique et sur le comportement du système en cas de collision.

Plusieurs types de méthodes existent pour contraindre le déplacement des objets. Nous n'expliciterons que la méthode choisie pour intégrer la contrainte de non collision. Nous expliciterons donc la méthode dite des pénalités et la résolution sous contraintes que nous avons mis en place.

Cependant, nous pouvons citer d'autres méthodes comme celle des déplacement imposés (utilisé notamment dans [70]) ou encore utilisant des pré-calculs pour contraindre le déplacement. La résolution de ces contraintes peut être effectuée par des méthodes dites pas à pas (à un instant donné, toutes les collisions sont traitées). Ce type de résolution peut ne pas être adapté à notre approche. En effet, il y a un temps de retard entre l'apparition de la contrainte et la résolution de celle-ci, pouvant entraîner des cas d'échec du suivi du mouvement (si la contrainte est résolue trop tard).

Méthodes de pénalité Dans le cas général, la collision entre deux objets ne doit pas exister. La manière naïve de répondre à une collision est d'utiliser une contrainte infinie empêchant tout contact entre 2 cônes de la chaîne articulaire. Cette approche se traduit

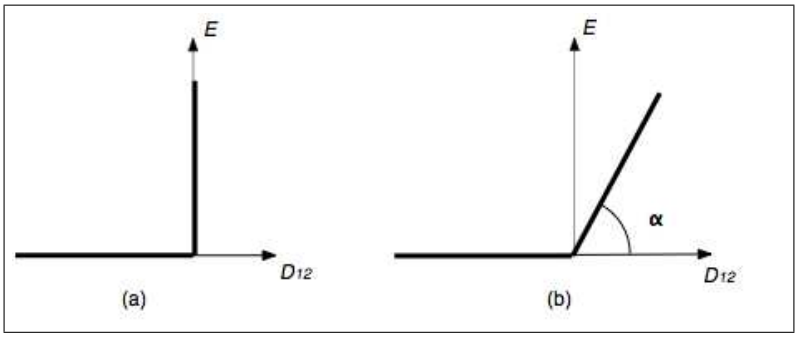


FIG. 7.10: La loi de Signorini (a) peut être régularisée par un facteur de pénalité α (b).

en pratique par une contrainte de type tout ou rien. Cette loi de contact est connue sous le nom de Loi de Signorini et est fortement non linéaire. Elle est peu pratique à utiliser dans le cadre de la minimisation standard où les fonctions de coût doivent être continues.

La méthode dite de pénalité consiste à introduire un facteur de régularisation dans la loi de Signorini. La loi devient alors continue (au moins par morceaux). Plus l'interpénétration est grande, plus la pénalité augmente. Dans la littérature, la fonction de pénalité est généralement linéaire par rapport à l'interpénétration.

$$E(D_{12}) = \begin{cases} \alpha D_{12} & \text{si} \quad D_{12} < 0 \\ 0 & \text{si} \quad D_{12} \geq 0 \end{cases} . \tag{7.11}$$

D_{12} peut être homogène à une distance (et ce sera notre cas) ([49]) ou à un volume ([68]). Ainsi, si D_{12} n'est pas nulle, la contrainte est proportionnelle à la distance d'interpénétration avec un coefficient α. Plus la valeur de α est élevée, plus on se rapproche d'une loi de contact idéale (loi de Signorini).

Cette loi est simple à mettre en place. Elle est en pratique très utilisée sous la forme présentée ci-dessus. Cependant, pour le rendu haptique, elle ne rend pas compte réellement du contact. Une seconde forme de pénalité est donc utilisée : elle permet de rendre compte du contact puis de la résistance de l'objet à la pression de l'utilisateur (c.f. figure 7.11-a).

Dans notre cas, nous voulons que la contrainte soit très faible pour une petite interpénétration pour laisser de la liberté à l'algorithme de minimisation. Cependant si l'interpénétration augmente de trop, alors il faut bloquer la minimisation. Cette contrainte peut être modélisée à l'aide de loi d'interaction de type Van der Waals. Cette force en $1/x^7$ rend compte de l'interaction entre les molécules. Elle augmente très rapidement pour les faibles distances, mais est très faible sinon. Nous pouvons transposer cette loi à notre problème de la manière suivante : si notre interpénétration est faible, la force de répulsion est faible. Arrivé à une certaine limite (L c.f. figure 7.11 (b)), la force augmente très vite pour éviter une trop grande interpénétration (et donc une perte du suivi du fait qu'une partie soit cachée dans l'autre). Cependant, il faut faire

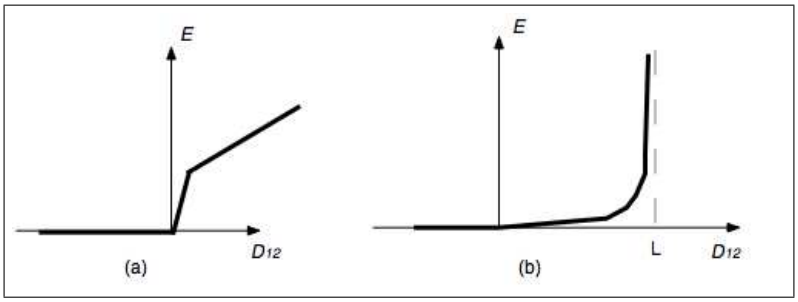


FIG. 7.11: Loi de résistance pour le rendu haptique (a). Loi utilisée pour la pénalité d'interpénétration (b).

attention à la vitesse de croissance de la pénalité. En effet, dans le cas où la croissance est trop forte, on se rapproche d'une contrainte binaire. Cette contrainte peut introduire alors de fortes discontinuités qui peuvent nuire lors de l'estimation des paramètres.

Résolution par minimisation sous contrainte Comme indiqué par le nom de la méthode, il s'agit de minimiser les forces de contact entre les objets de la scène. Si l'interpénétration entre les objets est nulle, alors il n'y pas de forces de contact, le système est à l'équilibre. Dès qu'une interpénétration apparaît, l'équilibre est rompu et l'objectif est de le ramener à l'équilibre.

Dans le cadre du suivi du mouvement humain, les contraintes de non pénétration influent sur les valeurs angulaires estimées. Lorsqu'une contrainte de limite angulaire est violée, l'approche naïve consisterait à bloquer l'évolution des angles. Cette méthode amène à une solution non optimale ou à des instabilités numériques. En effet, si un angle est fixé à sa limite, un degré de liberté est perdu pendant la phase de minimisation. Nous pourrions avoir un effet du type illustré par la figure 7.12-b. Le minimisation atteint un minimum local et ne converge pas vers la bonne solution du fait qu'un degré de liberté est bloqué (les deux articulations sont en collision au niveau de l'articulation).

Pour effectuer correctement la prise en compte des limites articulaires, il faut prendre en compte les contraintes lors la minimisation et directement dans l'incrément des valeurs articulaires. Posons $\boldsymbol{C}$ notre vecteur de contraintes et $\mathbf{J}_{\boldsymbol{C}}$ la Jacobienne associée. $\mathbf{J}_{\boldsymbol{C}}$ lie la variation des paramètres articulaires à la variation de la fonction de pénalité. Le problème de minimisation tel que présenté dans le chapitre 5 est modifié pour donner un problème de minimisation sous la contrainte $\boldsymbol{C}(\boldsymbol{\Phi}) = 0$. La résolution de ce problème peut se faire de la même manière que celle proposée. Cependant, l'incrément des variables articulaires (Φ_i) est modifié :

$$d\Phi = \mathbf{J}^{+}_{\boldsymbol{C}}\boldsymbol{C}(\boldsymbol{\Phi}) + (\mathbf{I} - \mathbf{J}^{+}_{\boldsymbol{C}}\mathbf{J}_{\boldsymbol{C}})d\Phi_0, \tag{7.12}$$

où $\mathbf{J}^{+}_{\boldsymbol{C}}$ est la matrice pseudo inverse de la Jacobienne $\mathbf{J}_{\boldsymbol{C}}$.

Cette équation peut être vue de la manière suivante :

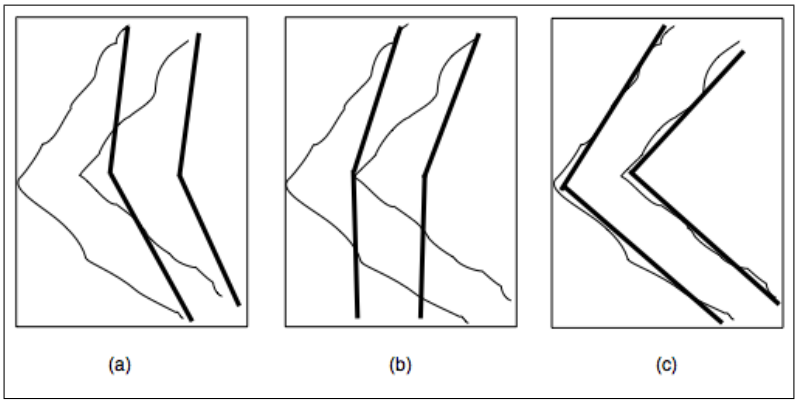


FIG. 7.12: Exemple simple de convergence erronée liée à une limitation dure des limites articulaires. La pose initiale du modèle (en gras) est modifiée pour correspondre aux contours observés dans les images (en clair). De la position initiale (a), on converge vers une solution non optimale en utilisant des contraintes articulaires dures (b). (c) représente la pose correcte.

1. Effectuer le nouvel incrément calculé (pour estimer la pose) uniquement sur les valeurs satisfaisant la contrainte $\boldsymbol{C}$.
2. Incrémenter les autres variables de la valeur corrigée pour que celle-ci satisfasse la contrainte.

Il s'agit de modifier les valeurs articulaires violant les contraintes de manière incrémentale pour qu'elles satisfassent toutes les contraintes.

Dans le cadre de nos travaux, nous avons mis en place une méthode de minimisation utilisant les fonctions de répulsion. D'autre part, nous intégrons ces pénalités dans le processus global d'estimation du mouvement. Nous allons maintenant expliciter le calcul de la Jacobienne de la pénalité.

7.5.2 La matrice Jacobienne de la fonction de pénalité

Nous avons vu dans le paragraphe 7.4.2 que la distance d'interpénétration est exprimée en fonction de la distance entre les axes principaux et les rayons des cônes. La distance entre les axes principaux dépend de la pose des deux cônes dans l'espace et donc des paramètres de la chaîne articulaire. Par conséquence, les rayons dépendent aussi des paramètres de pose. Il s'agit donc de déterminer la relation liant la variation de la distance d'interpénétration des cônes avec la variation des paramètres articulaires.

Nous allons dans un premier temps établir la dérivée de la distance entre les axes principaux. Puis nous calculerons la variation des rayons des cônes en fonction des paramètres articulaires.

Jacobienne de la distance Nous avons vu que la distance entre les axes principaux est de la forme : $\|\overrightarrow{P_1P_2}\|^2 = \| - s_c\boldsymbol{u} + \boldsymbol{w}_0 + t_c\boldsymbol{v}\|^2$, avec s_c, t_c, $\boldsymbol{w}_0$, $\boldsymbol{u}$ et $\boldsymbol{v}$ décrit

dans le paragraphe 7.4.1. Si nous notons α l'un des paramètres articulaires de la chaîne cinématique, alors la variation de la distance en fonction de la variation de ce paramètre s'écrit :

$$\frac{\partial}{\partial\alpha}\|\overrightarrow{P_1P_2}\| = \frac{\frac{\partial}{\partial\alpha}(-s_c\boldsymbol{u} + \boldsymbol{w}_0 + t_c\boldsymbol{v})^2}{2\|\overrightarrow{P_1P_2}\|}. \tag{7.13}$$

Remarque : Notons qu'il n'est pas nécessaire de calculer la variation de la distance par rapport aux paramètres du mouvement libre du modèle *3D*. En effet, le mouvement libre affecte l'ensemble des parties du corps de la même manière et donc la variation de la distance est nulle.

Pour calculer la variation de la distance, nous devons donc calculer :

$$\frac{\partial}{\partial\Phi}(-s_c\boldsymbol{u} + \boldsymbol{w}_0 + t_c\boldsymbol{v}) = -s_c\dot{\boldsymbol{u}} + \dot{\boldsymbol{w}}_0 + t_c\dot{\boldsymbol{v}} - \dot{s}_c\boldsymbol{u} + \dot{t}_c\boldsymbol{v}, \tag{7.14}$$

où par abus de notation, $\dot{x} = \frac{\partial}{\partial\Phi}x$. Nous allons maintenant nous attacher à calculer chacun des termes de cette équation. Nous pouvons remarquer que les dérivations de $\boldsymbol{u}$, $\boldsymbol{v}$ et $\boldsymbol{w}_0$ sont très similaires. Nous ne traiterons donc que de la dérivation de $\boldsymbol{u}$. D'autre part, t_c et s_c sont des fonctions de $\boldsymbol{u}$, $\boldsymbol{v}$ et $\boldsymbol{w}_0$. Leur dérivation est donc simple à effectuer si nous connaissons la dérivée de $\boldsymbol{u}$, $\boldsymbol{v}$ et $\boldsymbol{w}_0$.

Nous avons vu que les coordonnées de tous les vecteurs sont exprimées dans le repère de référence (celui du monde). En supposant que le cône i ait pour hauteur l_i, et que son axe principal soit aligné avec l'axe $\boldsymbol{k}$ du repère associé, alors le vecteur $\boldsymbol{u}_i$ associé à pour coordonnées (dans le repère du monde) :

$$\overline{\boldsymbol{u}}_i = \mathbf{D}(\boldsymbol{\Gamma})\mathbf{K}(\boldsymbol{\Lambda}_i)(0, 0, l_i, 0)^\top, \tag{7.15}$$

où $\overline{\boldsymbol{u}}$ est le vecteur de coordonnées homogènes associé à $\boldsymbol{u}$. Avec les résultats du chapitre 3, la dérivation de $\boldsymbol{u}$ vient facilement. Cependant, nous avons calculé la Jacobienne de la chaîne cinématique dans le cadre de la modélisation en référence zéro. Ce qui est facilement adaptable ici, en supposant que le vecteur des paramètres de pose initiale de la chaîne articulaire est le vecteur nul (dans leur position initiale, les repères des cônes sont confondus avec le repère de référence).

Nous avons donc :

$$d\boldsymbol{u}_i = \begin{bmatrix} -[\boldsymbol{u}_i]_\times & \mathbf{0}_{3\times3} \end{bmatrix} \mathbf{J}_{H_i} d\boldsymbol{\Phi}. \tag{7.16}$$

Le même calcul peut être effectué pour $\boldsymbol{v}$ et $\boldsymbol{w}_0$. On peut donc écrire la Jacobienne de la distance entre les axes principaux sous la forme d'une somme de cinq termes de sorte que :

$$d\|\overrightarrow{P_1P_2}\| = \mathbf{J}_{\overrightarrow{P_1P_2}} d\boldsymbol{\Phi}. \tag{7.17}$$

Jacobienne des rayons Nous allons maintenant expliciter la variation des rayons des cônes aux points P_1 et P_2 en fonction de la variation des paramètres articulaires.

Nous avons vu que le rayon du cône est de la forme :

$$R_i^2 = (a^2(1 + kz_i)^2\cos(\theta_i)^2 + b^2(1 + kz_i)\sin(\theta_i)^2, \tag{7.18}$$

où z_i et θ_i dépendent des paramètres de pose. Pour déterminer la variation de R_i, nous devons donc déterminer la variation de z_i, $\cos(\theta_i)$ et $\sin(\theta_i)$ en fonction des paramètres articulaires. Or z_i ne dépend que de t_c ou s_c. D'autre part, nous avons déterminé la forme explicite de $\cos(\theta_i)$ et $\sin(\theta_i)$, et la dérivée de ces deux termes est évidente à calculer en utilisant tous les résultats précédents.

Synthèse Les résultats précédents nous permettent de calculer la variation de la distance d'interpénétration des cônes. Elle dépend de manière explicite de la variation des paramètres articulaires. Cependant, comme nous avons pu le voir, il est assez complexe à calculer dans le sens où il y a beaucoup d'étapes dans les calculs pour estimer la variation des rayons.

Nous pouvons donc nous poser la question de la simplification des calculs de la Jacobienne. En effet, la pénalité est censée empêcher deux cônes de s'interpénétrer. La détection de la collision est effectuée à chaque pas de la minimisation et donc pour des incréments angulaires très faibles. Les corrections angulaires à apporter en cas de collision peuvent être relativement faibles, et par conséquence les déplacements relativement faibles. Nous pouvons donc nous demander l'intérêt d'étudier la variation du rayon des cônes pour des déplacements faibles. Nous aborderons cette étude dans le paragraphe suivant.

7.6 Extension

Nous avons pu voir dans la partie introductive de ce chapitre qu'il existait différentes méthodes pour prendre en compte les collisions. Nous avons choisi d'utiliser une méthode statique de détection des collisions. Ce choix est motivé par le fait que nous n'ayons pas de modèle de mouvement pour les différentes parties du corps. Cependant, nous pourrions utiliser une méthode de détection dynamique pour des parties du corps comme les mains et les pieds. Nous pourrions envisager un suivi temporel de ces membres et donc une connaissance du modèle du mouvement associé. En pratique, nous avons mis en place une méthode de suivi utilisant le filtrage particulaire pour les mains et les pieds (c.f. chapitre 8). Nous avons donc un modèle de mouvement mis à jour qui nous permettrait de calculer les positions des mains jugées acceptables en prenant en compte les collisions. En effet, dans la version simple présentée dans l'annexe, les collisions ne sont pas prises en compte, ce qui pour des situations où les mains sont proches d'une autre partie du corps pose des difficultés.

Chapitre 8

Suivi des pieds, des mains et de la tête

Sommaire

Résumé

Pour effectuer le suivi spécifique des pieds, des mains et de la tête, difficiles à suivre avec la méthode utilisant les contours, nous proposons d'utiliser une méthode de type filtrage particulaire. Ce suivi a pour objectif de contraindre le déplacement du modèle *3D* au cours de l'estimation du mouvement à l'aide des contours.

Nous adoptons une méthode de filtrage particulaire avec ré-échantillonnage adaptatif. Chacun des membres est suivi à l'aide d'un filtre particulaire. Ce choix pose des difficultés auxquelles nous proposons des solutions adéquates. D'une part, le mouvement humain ne peut pas réellement être caractérisé par un modèle de mouvement précis (position, vitesse ou accélération constante). Nous adoptons donc une approche avec un modèle de mouvement à deux états. D'autre part, dans le cas où deux membres du corps sont trop proches (comme les pieds lors de certaines phases de la marche), les particules associées à chacun des membres interagissent pour éviter de suivre le mauvais membre.

En sortie du filtrage, nous obtenons la position en *3D* de chacun des membres. C'est cette mesure que nous utilisons pour contraindre le déplacement du modèle *3D*. Nous proposons alors une fonction d'erreur rendant compte de la distance entre l'estimation *3D* et la position du cône attaché à chaque membre. Nous intégrons cette fonction dans la méthode globale de minimisation.

Enfin, nous donnons le résultat du filtrage sur une séquence complexe de marche et proposons des perspectives pour l'amélioration du filtrage.

Introduction au chapitre

Lors de l'estimation du mouvement à l'aide des contours, nous avons pu voir que certaines parties du corps pouvaient être mal détectées dans les images, que ce soit de par leur nature géométrique ou par leur petite taille dans les images.

La détection des mains, des pieds et de la tête dans les images pose notamment des difficultés. La forte variabilité de la géométrie de la main rend difficile la détection des contours et notamment la modélisation géométrique de celle-ci. Concernant les pieds, la difficulté est liée à la présence d'ombres à proximité et donc la présence de contours perturbants lors de la minimisation. Enfin, la tête apparaît avec beaucoup de contours distrayants (yeux, nez, bouche, lunettes, barbe et cheveux). Pour aider le suivi de ces différentes parties du corps, nous avons décidé de mettre en place une méthode de suivi spécifique.

Nous avons choisi d'utiliser une méthode de suivi par filtrage particulaire. Dans ce chapitre, nous allons dans un premier temps donner un état de l'art sur le filtrage en se concentrant sur des méthodes adaptées à notre problème. Puis nous justifierons le choix du filtrage particulaire pour effectuer le suivi des mains, pieds et tête. Nous montrerons comment nous l'avons adapté pour effectuer le suivi des différentes cibles et comment nous pourrons l'utiliser dans le cadre de la minimisation que nous avons introduite pour effectuer le suivi du mouvement (c.f. chapitre 5).

8.1 Etat de l'art et rappels

Pour effectuer le suivi d'une cible, que ce soit en *2D* dans les images ou en *3D* dans l'espace, il existe différentes techniques. Nous n'aborderons pas l'ensemble de ces méthodes dans le cadre de cette thèse.

Le problème du suivi d'une cible dans l'image est un problème très étudié. Les méthodes employées sont souvent spécifiques aux objets suivis. Parmi les techniques disponibles nous pouvons citer les méthodes de filtrage. Etant donné la mesure $\boldsymbol{z}_k$ à l'instant k et l'état du système $\boldsymbol{x}_{k-1}$ à l'instant $k-1$, l'objectif est d'estimer $p(\boldsymbol{x}_k|\boldsymbol{z}_1 \dots \boldsymbol{z}_k)$. Pour estimer l'état du système, nous utilisons les deux équations suivantes :

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \quad \text{qui est l'équation modélisant la dynamique du système} \tag{8.1}$$

$$p(\boldsymbol{z}_k|\boldsymbol{x}_k) \quad \text{qui est l'équation de la vraisemblance de l'estimation} \tag{8.2}$$

Rq : L'ensemble des états possibles $\boldsymbol{x}_k$ du système représente l'espace d'état.

Différentes méthodes de filtrage ont été proposées selon la linéarité ou non de ces équations. Lorsque les deux équations sont linéaires, le filtrage de Kalman ([150]) propose une solution optimale au problème. Dans le cas non linéaire, il existe plusieurs méthodes. Le filtre de Kalman étendu ou le filtre de Kalman *Unscented* sont des solutions sous-optimales dans le cas où le système est non linéaire, à la condition que la distribution soit uni-modale.

Pour palier à la non-optimalité des extensions du filtre de Kalman dans le cas des systèmes multi-modaux, d'autres approches, appelées méthodes par grille, proposent la construction d'un maillage déterministe de l'espace d'état. Cependant, il s'agit de méthodes complexes et leur utilisation peuvent devenir problématique dans le cadre d'espace de grande dimension (> 4). L'utilisation de méthodes séquentielles de Monte Carlo constitue une alternative facile à mettre en oeuvre à ces méthodes.

Nous utilisons le filtrage séquentiel de Monte Carlo encore appelé le filtrage particulaire. Beaucoup de méthodes de ce type ou dérivées ont été développées. Pour plus de détails sur le filtrage séquentiel, le lecteur pourra se référer à [44]. Enfin, nous adopterons les notations du tutoriel [7], qui avec la thèse [6] (en français) proposent un bon état de l'art récent du domaine.

Dans la suite de ce paragraphe, nous expliciterons l'algorithme de filtrage dit BootStrap ([58]) avec ré-échantillonnage adaptatif et multimodal que nous utilisons.

8.1.1 Le filtrage particulaire

Nous abordons maintenant le principe du filtrage particulaire. Prenons l'objectif simple de suivre une cible dans une image. Nous supposons que nous connaissons la cible, c'est-à-dire que nous avons un modèle de celle-ci. Supposons connu la position de la cible à l'instant t. Nous voulons estimer la position de la cible à l'instant $t+1$. Pour cela, nous connaissons le modèle de déplacement de la cible mais avec une certaine incertitude. Pour modéliser cette incertitude, nous générons plusieurs déplacement de cible en effectuant un tirage aléatoire sur la position de la cible (ce déplacement correspond à $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$). Nous créons ainsi un nuage de position à l'instant $t+1$. Parmi ces nouvelles positions certaines sont correctes et d'autre non. Pour estimer la position de la cible, il faut calculer la vraisemblance de chacune des nouvelles positions avec la mesure ($p(\boldsymbol{x}_k|\boldsymbol{z}_k)$). Au final, chaque position est dotée d'un poids caractérisant la correction de la position ou non. Nous appellerons *particule* une position. Chaque particule à un poids associé qui dépend de la vraisemblance. Une particule est donc une hypothèse de l'espace d'état.

L'estimation de la position de la cible (la distribution objectif) est approchée par une somme pondérée discrète, dont le support correspond aux positions et les coefficients aux poids. Enfin, pour éviter d'effectuer le suivi avec des particules ayant une mauvaise vraisemblance, nous ré-échantillonnons le nuage de particules pour éliminer les particules de plus faible poids.

8.1.2 Notre filtrage particulaire

Nous avons décrit le principe général du filtrage particulaire, nous allons maintenant nous attarder sur la méthode que nous utilisons en pratique. Comme nous l'avons vu plus haut, nous effectuons le suivi de plusieurs cibles (des pieds, des mains et de la tête d'un acteur). Pour cela, nous utilisons un filtre (composé de N particules) par cible suivie. Dans la suite de ce paragraphe, nous nous restreignons à un filtre unique (sauf mention contraire).

Pour effectuer le suivi, nous utilisons le filtre BootStrap avec échantillonnage adaptatif. Nous verrons cependant que le complexité du suivi nous a amené à considérer une méthode de ré-échantillonnage différente de celle proposée dans cet algorithme. Dans cet algorithme, N est le nombre de particules (pour un filtre), $\boldsymbol{x}_k^{(i)}$ la position dans l'espace d'état associé à la i$^{\text{e}}$ particule à l'instant k, $w_k^{(i)}$ le poids associé à la i^{e} particule et $\boldsymbol{z}_k$ est la mesure ou l'observation. Nous verrons l'interprétation et la signification de l'ESS_N au paragraphe 8.2.2.

De manière descriptive, l'algorithme 1 comprend une initialisation et 3 étapes.

L'initialisation : De même que pour le suivi utilisant les contours, nous avons une initialisation de la pose du modèle. Nous avons donc connaissance de la position des pieds, des mains et de la tête dans la première image de la séquence. Cette localisation nous permet d'initialiser le filtrage ainsi que le modèle de peau que nous utiliserons pour effectuer le calcul de la vraisemblance. Nous créons un filtre par partie du corps. Pour avoir une première distribution, nous appliquons un bruit sur l'ensemble des particules (*jittering*).

Propagation : Les particules de chacun des filtres sont propagées selon un modèle donné. Le modèle dynamique que nous avons choisi d'utiliser est composé d'un modèle de mouvement à deux états (position constante et vitesse constante) que nous aborderons au paragraphe 8.2.3. Enfin, pour modéliser l'incertitude de la nouvelle position, nous appliquons un bruit gaussien sur les nouvelles positions. Une fois les particules propagées, nous déterminons la cohérence de la particule avec l'observation, nous obtenons ainsi une mesure de vraisemblance par particule qui permet d'évaluer sa pertinence.

Estimation : Il s'agit de mettre en avant une particule représentative de la distribution au sein d'un filtre. De manière générale, deux choix sont possibles. Soit nous choisissons la particule ayant le poids le plus fort, soit nous faisons la moyenne pondérée de l'ensemble des particules. Nous choisissons la seconde solution.

Ré-échantillonnage : La répétition de l'étape de propagation et du calcul des poids conduit en général à une augmentation de la variance des poids dans le temps. En pratique, cela a pour effet de faire décroître rapidement le nombre de particules significatives dans le nuage. Ce problème de dégénérescence conduit à une divergence du nuage et donc à une détérioration de l'estimation. Il faut donc « resserrer » les particules autour de l'observation ou de la mesure correcte. Pour cela, nous effectuons un ré-échantillonnage des particules si besoin est. Pour déterminer la nécessité ou non du ré-échantillonnage, nous calculons la taille effective du N-échantillon (ESS_N : *effective sample size*). Si la valeur n'est pas au dessus d'un certain seuil, nous effectuons un ré-échantillonnage : nous sélectionnons ainsi plusieurs particules parmi celles qui ont un poids fort et régénérons un nuage de N particules à partir de celles-ci. Nous verrons la méthode en détail au paragraphe 8.2.2.

Algorithme 1 Filtre bootstrap avec ré-échantillonnage adaptatif

- **initialisation :**

  pour $i = 1...N$, générer $\boldsymbol{x}_0^{(i)} \sim p(\boldsymbol{x}_0)$, et fixer $w_0^{(i)} = 1/N$

pour $k = 1, 2, ...$, nous effectuons les étapes suivantes :

- **échantillonnage pondéré séquentiel :**
  1. échantillonnage : pour $i = 1...N$, générer $\boldsymbol{x}_k^{(i)} \sim p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}^{(i)})$
  2. mise à jour des poids d'importance : pour $i = 1...N$, calculer $w_k^{(i)} = w_{k-1}^{(i)} p(\boldsymbol{z}_k|\boldsymbol{x}_k^{(i)})$
  3. normalisation des poids : pour $i = 1...N$, calculer $\widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$
- **estimations de Monte Carlo**

$$\mathbb{E}[\phi(\boldsymbol{x}_k)] \simeq \sum_{i=1}^{N} \phi(\boldsymbol{x}_k^{(i)})\ \widetilde{w}_k^{(i)}$$

où ϕ est une fonction intégrable. Dans notre cas, nous avons :

$$\mathbb{E}[\boldsymbol{x}_k] = \sum_{i=1}^{N} \boldsymbol{x}_k^{(i)}\ \widetilde{w}_k^{(i)}$$

- **ré-échantillonnage adaptatif :**
  1. calculer le seuil de ré-échantillonnage :

$$ESS_N = \frac{1}{\sum_{j=1}^{N} \widetilde{w}_k^{(i)}}$$

  2. si $ESS_N < seuil$:
     - tirer avec remise N particules $\widetilde{\boldsymbol{x}}_k^{(i)}$ parmi $\{\boldsymbol{x}_k^{(i)}\}_{i=1...N}$ proportionnellement aux poids $\{\widetilde{w}_k^{(i)}\}_{i=1...N}$
     - pour $i = 1...N$, poser $\boldsymbol{x}_k^{(i)} = \widetilde{\boldsymbol{x}}_k^{(i)}$ et $\widetilde{w}_k^{(i)} = 1/N$

8.2 Application au suivi des membres

Dans le paragraphe précédent, nous avons décrit l'algorithme de filtrage que nous utilisons. Nous allons maintenant présenter la mise en œuvre pratique de l'algorithme pour le suivi des mains, pieds et tête de l'acteur. Ce filtrage nous permet d'aider la méthode d'estimation des paramètres de pose introduite dans les chapitres précédents, dans des cas où les contours extraits des images ne permettent pas d'estimer correctement la pose.

Nous sommes dans une situation où nous avons plusieurs caméras calibrées et synchronisées. Nous effectuons donc le suivi des cibles dans l'espace réel c'est-à-dire dans l'espace *3D*. Nous avons décidé de nous restreindre à l'estimation de la position des membres dans l'espace sans estimer leur orientation. En effet, l'estimation de l'orientation nécessite une résolution plus élevée que celle dont nous disposons.

Dans la suite de cette section, nous abordons dans un premier temps l'observation utilisée pour effectuer le calcul de la vraisemblance de chacune des particules. Puis, nous aborderons le choix du modèle de propagation de nos particules au cours du temps. Enfin, nous décrirons la méthode employée pour effectuer le ré-échantillonnage des particules.

8.2.1 La mesure

Nous voulons suivre les pieds, les mains et la tête de l'acteur. Pour effectuer le suivi, nous disposons du modèle *3D* de l'acteur ainsi que des images couleur. Plusieurs types d'observations peuvent être utilisés pour effectuer le suivi : les contours, des points d'intérêts, la couleur ou encore des modèles d'apparence. Nous avons décidé d'utiliser la couleur. En effet, les modèles de couleur sont persistants et robustes ([117], [118]).

Cependant, la détection de la peau pose de nombreux problèmes (variabilité des couleurs, réfléctance...) que nous n'avons pas abordé au cours de ces travaux. Beaucoup de travaux proposent des solutions pour détecter la peau de manière robuste dans les images. Pour un état de l'art récent, le lecteur pourra se référer à [151].

Pour effectuer la détection de la peau, nous pouvons nous placer dans différents espaces de couleur. Nous avons choisi d'utiliser l'espace HSV (c.f. figure 8.1-(b)). Le modèle de couleur de la peau est appris en s'aidant du modèle *3D*. Plus précisément, nous estimons la pose de l'acteur dans la première image de la séquence vidéo. Nous utilisons les contours projetés du modèle dans chacune des images pour délimiter une zone sur laquelle la couleur de la peau est apprise. Pour éviter des apprentissages erronés, nous prenons en compte la visibilité de ces membres dans les différentes caméras. Nous construisons alors un histogramme modèle de la peau pour chaque membre et dans chaque image. Cependant, la couleur de la peau peut varier au cours de la séquence vidéo. Pour prendre en compte plus de variabilité de la couleur de la peau, dans une image donnée, nous accumulons les histogrammes de chaque cible pour construire un modèle de couleur pour toute les cibles vues dans une image. Cet histogramme permet

de calculer la probabilité qu'un pixel de l'image soit un pixel de peau ou non (c.f. figure 8.1-(c)). Nous obtenons ainsi une carte de probabilité de la peau dans les images. Pour calculer la vraisemblance d'une particule, nous utilisons les cartes de probabilité de présence de la peau. Une particule décrit une position *3D* ainsi qu'une taille (caractéristique de la taille de la cible). La vraisemblance se mesurant dans les images, la particule est projetée sur chacune des carte de probabilité. Chaque particule décrit dans chacune des images une position *2D* et une région (c.f. illustration 8.2). Nous calculons la vraisemblance d'une particules dans une image est déterminant le taux de recouvrement $\tau_c^{(i)}$ des pixels de peau dans la région :

$$\tau_c^{(i)} = \frac{1}{w * h} \sum_{k=0}^{w} \sum_{l=0}^{h} I_c^{(i)}(k, l), \tag{8.3}$$

où $I_c^{(i)}$ est le *patch* associé à la particule i pour l'image c. w et h sont les dimensions du patch. $\tau^{(i)} = 1$ si tous les pixels du patch ont une probabilité de 1 d'être de la peau. Cette mesure nous donne la vraisemblance par rapport à une image. Pour une particule, sa position est cohérente avec l'observation, si la mesure dans chacune des images où la particule est visible est bonne. Pour connaître la vraisemblance d'une particule, il faut alors multiplier les taux de recouvrement pour chacune des images. Du fait de la description dynamique du système, le poids d'une particule dépend de la vraisemblance mais aussi de son passé. Nous avons donc :

$$w_k^{(i)} = w_{k-1}^{(i)} \prod_{l=0}^{N_c} \tau_c^{(i)}, \tag{8.4}$$

où N_c est le nombre d'images. Le fait qu'une particule soit invisible dans une image ne doit pas pénaliser la vraisemblance, nous avons donc choisit de poser $\tau_c^{(i)} = 1$ le cas échéant. Cette équation n'est rien d'autre que $p(\boldsymbol{z}_k|\boldsymbol{x}_k)$.

8.2.2 Le ré-échantillonnage des particules

Pour que le suivi d'une cible et donc le filtrage s'effectue de manière correcte, il faut éviter une exploration trop importante et inutile de l'espace d'état. En effet, l'exploration trop grande implique la présence de mesures erronées et donc perturbantes pour l'estimation de la position. Nous éliminons donc les particules de poids le plus faible, l'objectif étant de recentrer le nuage de particules autour de la bonne estimation. Pour éviter d'effectuer cette étape à toutes les itérations, nous effectuons cette étape de ré-échantillonnage (*resampling*) à la condition que l'ESS_N soit plus faible qu'un seuil donné. D'autre part, la sélection des bonnes particules ne doit pas se limiter aux seules particules de poids élevé. Si nous effectuons une telle sélection, nous perdons toute l'information spatiale introduite par le filtrage. Nous effectuons donc un échantillonnage adaptatif et aléatoire de notre nuage de particules.

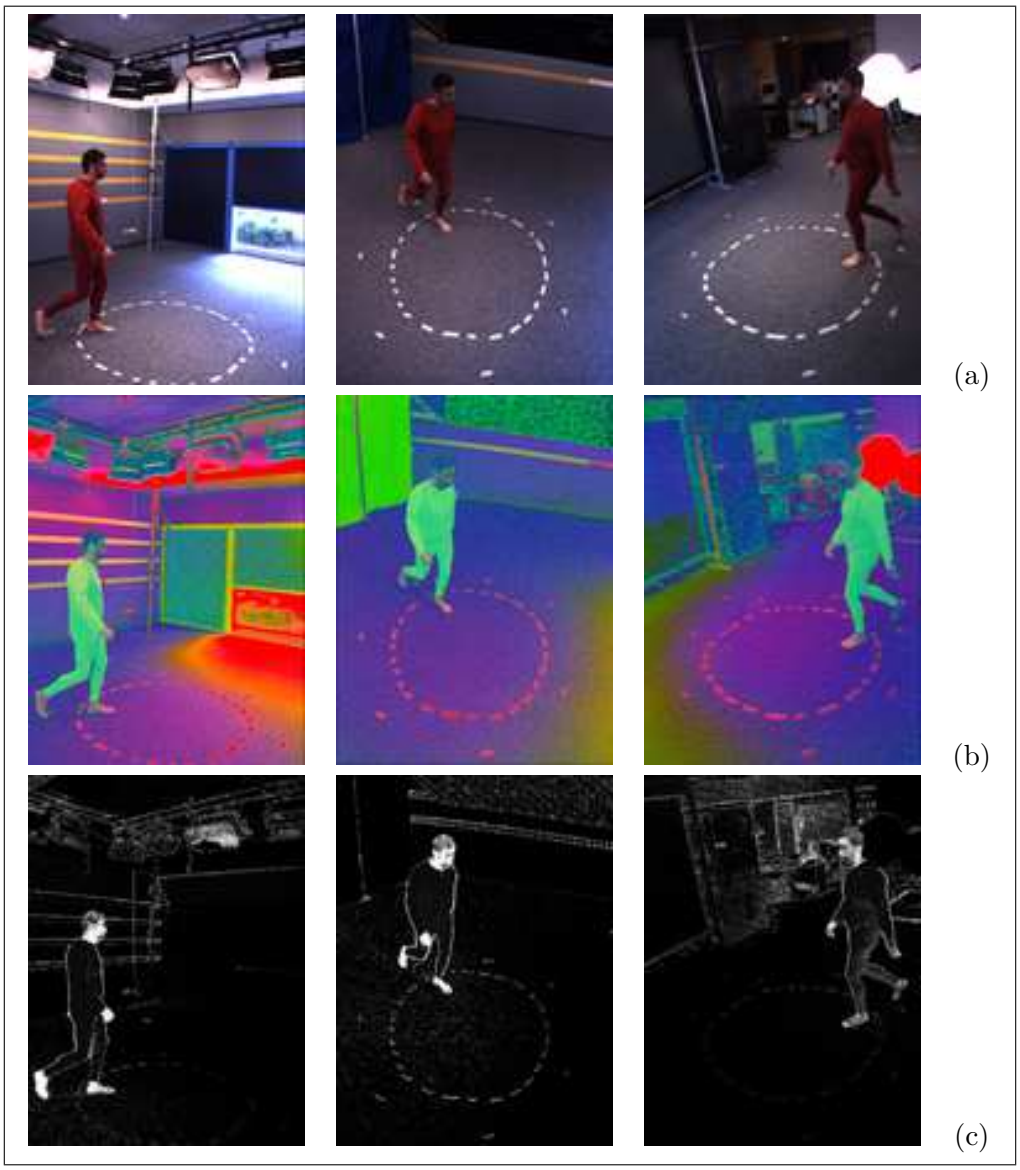


FIG. 8.1: Pour effectuer la détection de la peau, nous nous plaçons dans l'espace HSV (b). Nous créons alors un modèle (histogramme) de peau par image. L'histogramme permet de calculer une carte de probabilité de présence de la peau dans les images (c).

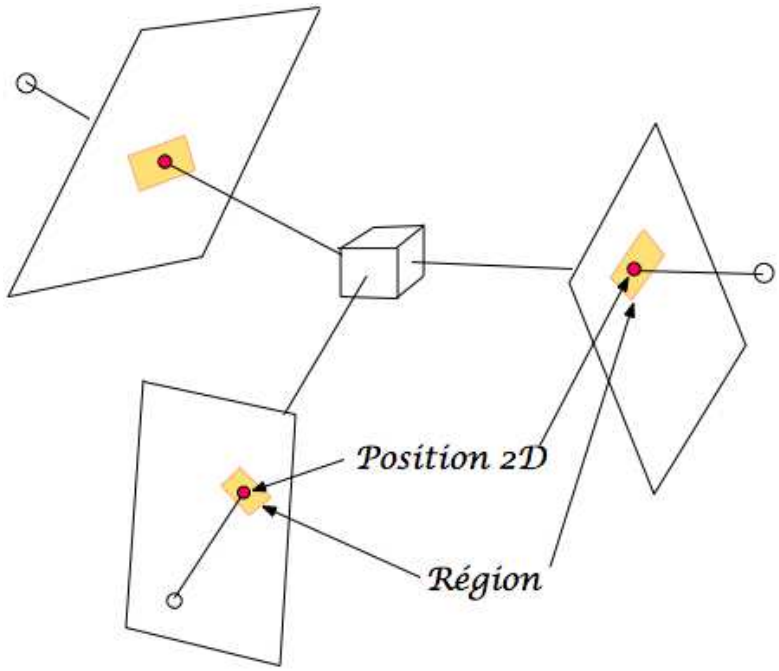


FIG. 8.2: Une particule *3D* se projette dans les images et décrit dans chacune d'elle une région.

La taille effective du N-échantillon La répétition des étapes de propagation et du calcul des poids (donné avec l'équation (8.4)) conduit à une augmentation de la variance du nuage de particules dans le temps. En pratique, cela a pour effet de faire décroître rapidement le nombre de particules significatives. Ce problème de dégénérescence conduit à une divergence du nuage et donc à une détérioration de l'estimation. Une mesure de cette dégénérescence a été proposée dans [94]. Nous pouvons approcher la valeur de cette mesure par l'expression suivante :

$$ESS_N = \frac{1}{\sum_{j=1}^{N} \widetilde{w}_k^{(i)}}. \tag{8.5}$$

L'ESS_N est la taille effective du N-échantillon (*effective sample size*) et correspond en pratique à l'inverse de la variance des poids des particules d'un filtre. Plus l'ESS_N est faible, plus le nombre de particules ayant un poids élevé par rapport aux autres est faible. Si toutes les particules ont le même poids, l'ESS_N est très élevé. Si toutes les particules ont un poids élevé alors ce cas est favorable. Cependant, un ESS_N élevé peut aussi être dû au fait que toutes les particules aient un poids faible, ce qui n'est pas une situation favorable. L'ESS_N permet de caractériser la distribution des particules et donc d'estimer le besoin d'effectuer un ré-échantillonnage.

La sélection des particules Pour effectuer le ré-échantillonnage du nuage de particules, plusieurs possibilités ont été proposées. Nous pouvons citer le ré-échantillonnage multinomial ([94]), le ré-échantillonnage résiduel ([95]). Enfin, nous utilisons le ré-échantillonnage systématique[1] introduit dans [23].

En pratique, il s'agit de sélectionner quelques particules parmi les N tout en privilégiant celles qui ont une vraisemblance élevée. Pour effectuer cette sélection, nous construisons une fonction strictement croissante qui est la somme des poids de toutes les particules (c.f. figure 8.3). Il s'agit ensuite d'échantillonner cette fonction pour sélectionner les particules. Pour cela, nous tirons une valeur u_0 selon une loi normale. Puis nous construisons la suite :

$$u_i = u_0 + i/N, \tag{8.6}$$

avec $u_0 = \mathcal{N}(0,1)/N$. Pour chaque valeur de u_i, nous gardons la particule associée (c.f. figure 8.3). Nous obtenons ainsi un échantillon de $N^{'} < N$ particules. Pour régénérer le nuage de N particules, nous rajoutons les particules manquantes en dupliquant certaines de celles échantillonnées.

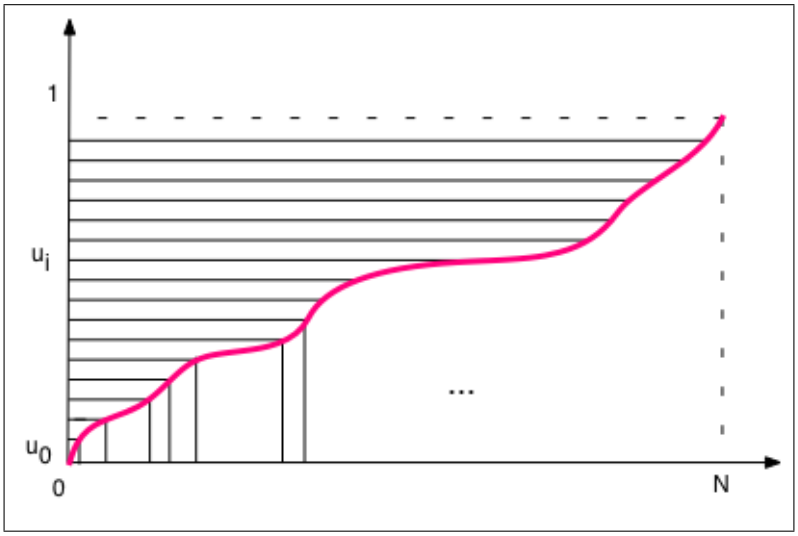


FIG. 8.3: Pour effectuer l'échantillonnage des particules, nous construisons une fonction strictement croissante, somme des poids des particules. Le tirage des particules s'effectue ensuite en sélectionnant les particules à incrément régulier sur le poids des particules.

8.2.3 Le modèle de mouvement

Pour effectuer le suivi d'une cible entre deux images consécutives, nous faisons évoluer chacune des particules selon un modèle de déplacement. Le modèle de

[1]Le terme "systematic resampling" est utilisé pour définir une méthode de ré-échantillonnage, à ne pas confondre avec le fait de réaliser une étape de ré-échantillonnage à chaque pas de temps.

déplacement dépend de l'application choisie. Nous pouvons citer des modèles génériques de type position, vitesse ou encore accélération constante.

Pour modéliser le déplacement des pieds, des mains et de la tête, nous ne disposons pas de mesure a priori. Nous utilisons donc un modèle mixte de mouvement alternant de manière quasi-aléatoire entre le mouvement à position constante et le mouvement à vitesse constante.

Ce choix est motivé par la complexité du suivi des cibles dans les images. Prenons l'exemple de la marche et plus précisément du mouvement des pieds. Du point de vue du référentiel du monde, le mouvement du pied a deux états : l'un est à position constante, si le pied est au sol, et l'autre est à vitesse (ou accélération) constante lorsque celui-ci est « décollé » du sol. Le choix d'un modèle de mouvement avec deux états est donc justifié.

En pratique, nous disposons de deux modèles de mouvements pour chaque particule. Lors de l'initialisation, le modèle de mouvement est tiré de manière aléatoire pour chaque particule. Nous avons donc deux ensembles de particules, les unes avec un modèle à position constante et les autres avec un modèle à vitesse constante. Pour une particule donnée, le passage d'un modèle à l'autre n'est possible que sous certaines conditions. D'une part nous contraignons la particule qui doit garder un modèle de mouvement pendant une période de temps donné. D'autre part, le changement de modèle n'est possible que si la vraisemblance de la particule n'est pas bonne. Cette alternance permet de prendre en compte de manière simple les deux états de mouvement. Dans le cadre de la marche, cette méthode s'avère efficace en pratique (c.f. figure 8.7).

Le choix de l'alternance peut cependant amener à des situations où les particules à position constante et les particules à vitesse constante sont cohérentes avec la mesure (par exemple certaines particules attachées au pied posé par terre et qui ont donc le modèle à position constante ; et les autres particules du filtre attachées au pied qui avance qui avance et qui ont le modèle à vitesse constante). Nous pouvons donc voir apparaître deux modes dans la distribution spatiale des particules. Dans ce cas, l'étape d'estimation de Monte Carlo doit prendre en compte cet aspect bi-modal.

8.2.4 Le filtrage bi-modal

Ce que nous allons présenter maintenant s'inspire du travail proposé dans [148].

Lorsque nous effectuons le suivi de plusieurs cibles dans une séquence vidéo, il peut arriver que deux cibles se croisent ou encore que du bruit lors de la détection de la peau perturbe le filtrage particulaire. Le cas le plus explicite est celui de la marche où les deux pieds se croisent souvent au cours du mouvement. Comme nous n'avons pas de modèles distinctifs des pieds, les particules d'un pied peuvent estimer la mesure sur l'autre pied correcte et donc rester « attachées » à ce pied. Il peut en résulter la perte de la cible à suivre et le fait que deux filtres soient attachés à la même mesure (c.f. figure 8.5-(a)). Pour palier à ce problème, nous avons mis en place un filtrage qui a la possibilité de passer dans un mode multi-modal. Ce mode permet à un filtre, si nécessaire, de suivre

deux cibles simultanément. Nous allons maintenant expliquer comment nous détectons le changement de mode, puis comment nous procédons au suivi multi-cibles.

Pour expliciter le propos, nous allons nous restreindre à deux cibles qui ont une apparence similaire chacune étant suivie avec un filtre. Lorsque les deux cibles sont suffisamment éloignées l'une de l'autre, les deux filtres agissent de manière indépendante. La difficulté apparaît lorsque les deux cibles se rapprochent l'une de l'autre. Les filtres peuvent s'attacher à n'importe quelle cible (puisque leur apparence est similaire). Lorsque les cibles s'éloignent l'une de l'autre, il faut s'assurer que les filtres continuent à suivre les deux cibles. En pratique, lorsque les cibles s'éloignent l'une de l'autre, certaines particules de chacun des filtres vont avoir tendance à s'attacher à la cible s'éloignant. Chacun des nuages de particules vont donc avoir tendance à s'étendre tout en gardant un ESS_N élevé (puisque la mesure est correcte). C'est cette augmentation de la variance du nuage qui nous permet de passer dans un mode bi-modal pour le ré-échantillonnage.

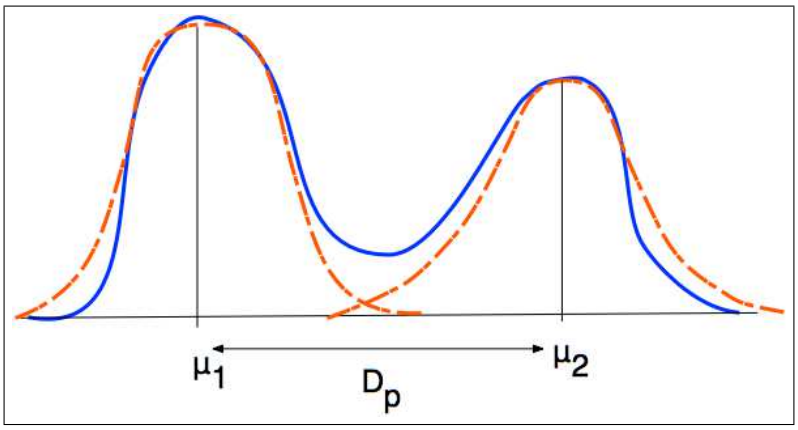


FIG. 8.4: Nous représentons la distribution des particules. Pour détecter l'aspect bi-modal du filtre, nous regardons la distance D_p séparant les centres des gaussiennes ajustées sur la distribution à l'aide de l'algorithme EM.

Pour détecter l'aspect bi-modal, nous utilisons un algorithme de type EM pour voir si le nuage particules peut être modélisé avec deux gaussiennes dont les moyennes sont suffisamment éloignées. Le cas échéant, le filtre est passé en mode bi-modal. Nous pouvons aussi utiliser des critères de type MDL (*Minimim Description Length* [63]) ou encore BIC (*Bayesian Information Criterion* [127]), pour estimer le nombre de gaussiennes modélisant la distribution spatiale des particules. Nous nous limitons ici au cas de deux cibles qui se croisent. Pour désigner l'ensemble des particules les plus proches d'un des centre d'une gaussienne, nous parlerons de **protos**. Nous obtenons donc deux ensembles de particules, chacune des particules étant associée à une cible. Un filtre est donc attaché à deux cibles.

Si nous détectons deux modes bien distincts, nous modifions l'algorithme BootStrap avec ré-échantillonnage adaptatif pour que ce dernier soit multi-modal. Au lieu de ré-échantillonner l'ensemble des particules d'un seul coups, nous adaptons le ré-échantillonnage pour ré-échantillonner chacun des protos de manière indépendante. Ce

ré-échantillonnage permet d'équilibrer le nombre de particules associées à chaque cible. Si nous avons 200 particules pour un filtre, alors chaque protos aura 100 particules. Pour effectuer ce ré-échantillonnage, la seule modification à apporter par rapport à l'algorithme de ré-échantillonnage que nous utilisons dans le cas général est de poser $u_0 = 1/N'$ où N' est le nombre de particules désiré par protos. Chacun des protos devient indépendant et un filtre peut alors suivre deux cibles.

Prise de décision : Si nous revenons au cas du suivi de l'ensemble des membres, chaque filtre a la possibilité de passer en mode bi-modal. Ce passage peut être justifié ou non. En effet, l'algorithme EM peut autoriser le passage au modèle bi-modal si une détection erronée entraîne des particules loin de la mesure réelle. D'autre part, il n'est pas nécessaire qu'un filtre se scinde en deux pour qu'un des protos s'attache à un membre déjà suivi par un autre filtre. Nous avons donc mis en place deux critères heuristiques pour éviter de garder des aspects bi-modaux dans des situations inutiles :

- si le poids moyen d'un des protos est très faible comparé à celui du second protos, nous regroupons les deux protos sur celui de poids le plus fort. Cette décision empêche de scinder un filtre suite à une détection erronée.
- si un protos d'un filtre scindé en deux se trouve à proximité d'un autre filtre, nous pouvons arrêter le suivi à l'aide de ce protos et ré-échantilloné le second protos pour avoir N particules sur ce dernier. Cette seconde décision permet de ne pas suivre deux fois la même cible.

8.2.5 Résultats

Nous allons présenter les résultats du suivi de cibles à l'aide du filtrage particulaire que nous avons adapté au problème du suivi des mains, des pieds et de la tête.

La détection de la peau La figure 8.1 présente des résultats de détection de la peau à l'instant initial de la séquence vidéo. Nous pouvons observer que des éléments autre que les membres de couleur peau sont détectés dans les images. Nous pouvons voir apparaître les cartons et les luminaires. Ces détections sont liées au fait que leurs couleurs sont très proches de celle de la peau. Ces détections rendent ambiguës la mesure de la vraisemblance. Cependant, les membres apparaissent de manière plus marquée dans ces images. L'utilisation de plusieurs points de vues pour la calcul de la vraisemblance d'une particule aide aussi à désambiguïser la mesure.

Le suivi Avec la figure 8.5, nous illustrons l'intérêt du filtrage multimodal que nous proposons. Dans cette série d'illustrations, la main gauche n'est pas suivie au cours de la séquence. En effet, cette main n'est pas visible dans la phase d'initialisation du filtre (c.f. figure 8.1).

La première ligne de la figure présente les résultats du suivi en utilisant le filtrage particulaire avec ré-échantillonnage unimodal. Nous pouvons voir que le pied droit n'est plus suivi après son passage à proximité du pied gauche. La seconde ligne illustre le

ré-échantillonnage multimodal. Sur la deuxième image nous pouvons voir que le filtre associé à la main est scindé en deux protos. Ils sont apparus du fait d'une mesure erronée. A la troisième image, nous pouvons voir que le filtre associé au pied droit se scinde en deux protos. Ces deux protos apparaissent du fait que les pieds se croisent. Ces deux protos évoluent avec chacun des pieds de manière indépendante. La dernière ligne illustre l'algorithme complet avec la prise de décision. Nous pouvons voir que sur la seconde image, l'algorithme a décidé que l'un des protos de la main était lié à une erreur de mesure. Sur la troisième image, le protos associé au pied gauche est détruit, car un filtre est déjà présent sur ce pied.

La figure 8.6 illustre les position *3D* estimée au cours de la séquence de marche. Enfin, la figure 8.7 illustre la séquence de marche.

8.3 Future intégration et perspectives

Dans ce paragraphe, nous présentons dans un premier temps la méthode que nous préconisons (mais qui dans l'état actuel n'est pas testée) pour intégrer le filtre particulaire comme aide au suivi du mouvement basé contours. Puis, nous abordons quelques perspectives qui permettraient de rendre notre suivi à l'aide du filtrage particulaire plus robuste.

8.3.1 Future intégration

Nous avons mis en place le filtrage particulaire des pieds, des mains et de la tête pour aider le suivi introduit au chapitre 5. L'objectif est donc de contraindre le suivi du mouvement utilisant les contours pour que les cônes associés aux mains, pieds et tête soient correctement positionnés. Nous avons décidé d'intégrer le filtrage particulaire comme une contrainte sur l'estimation des paramètres de la chaîne cinématique. Plus précisément, nous ajoutons dans la fonction de coût à minimiser une pénalité qui augmente la position *3D* des pieds, des mains ou de la tête s'écartent de l'estimation effectuée à l'aide du filtrage particulaire.

La fonction d'objectif La nouvelle fonction d'objectif est :

$$E_{Tot} = E + E_{Part}, \tag{8.7}$$

où E représente la fonction de coût introduite au chapitre 5. Nous allons maintenant définir E_{Part} qui est la fonction de pénalité associée au filtrage particulaire.

Nous avons décidé de définir E_{Part} de la manière suivante :

$$E_{Part} = \alpha_{Part} \sum_{f=0}^{N_F} \|\boldsymbol{O}_f - \mathbb{E}_f\|^2, \tag{8.8}$$

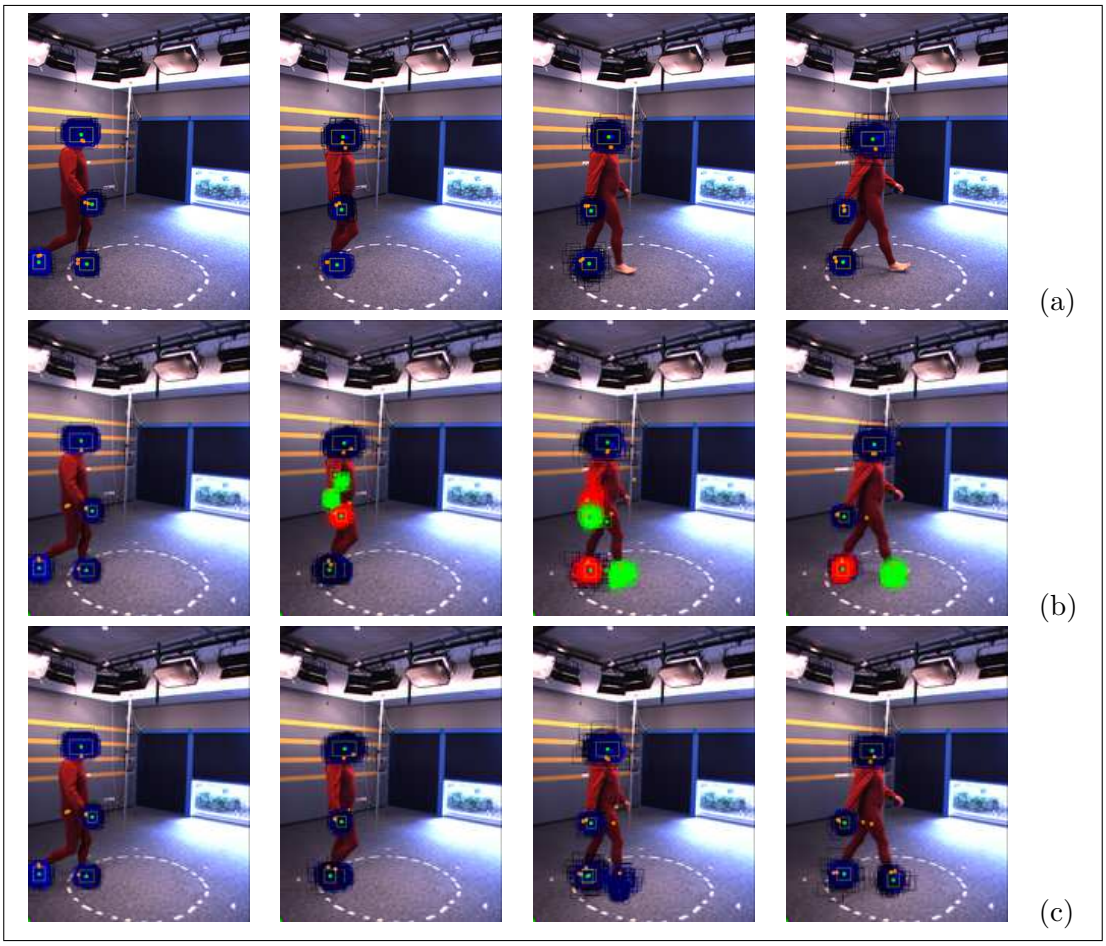


FIG. 8.5: Nous illustrons ici le suivi à l'aide du filtrage particulaire pour différentes étapes de l'algorithme. La première ligne montre l'échec du filtrage lors de l'approche naïve ne prenant pas en compte le ré-échantillonnage multi-modal. La seconde ligne présente le suivi lorsque nous utilisons le ré-échantillonnage multi-modal. Les blobs verts et rouges sont une représentation des modes de chacun des filtres. Les blobs bleus ne sont pas multimodaux. Enfin, la dernière ligne représente le filtrage lorsque la prise de décisions de refusion des modes est mise en place.

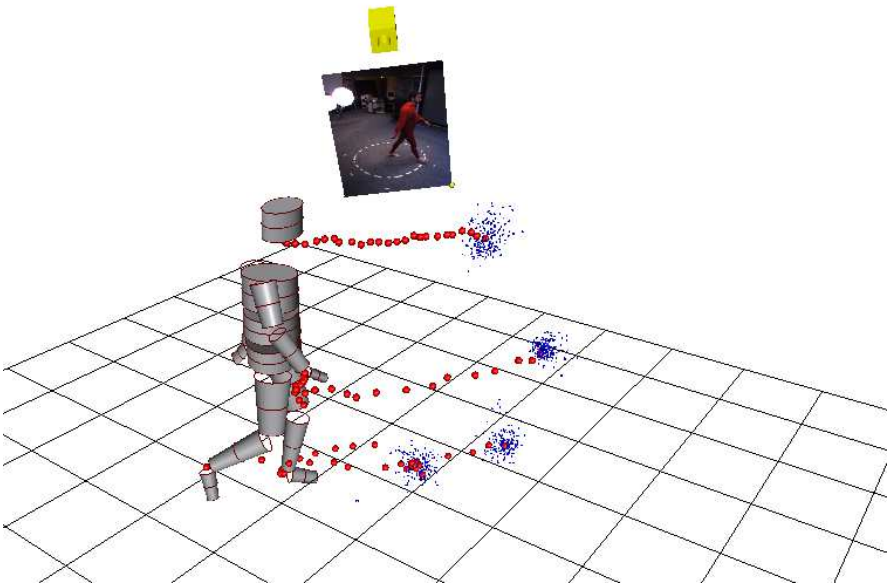


FIG. 8.6: Positions successives des particules au cours des vingt premières images de la séquence de marche.

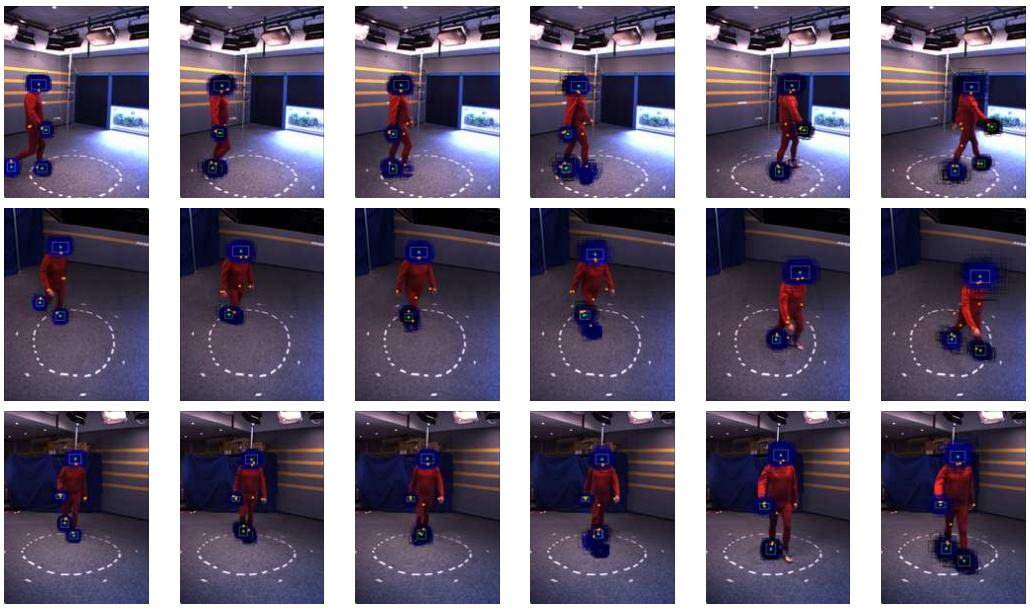


FIG. 8.7: Le suivi s'effectue en *3D*. Nous présentons ici plusieurs points de vue du suivi par filtrage pour la séquence de marche.

où α_{Part} est un coefficient permettant de donner plus ou moins d'importance à la pénalité, N_F est le nombre de filtres (dans notre cas cinq : deux pieds, deux mains et une tête), $\boldsymbol{O}_f$ est le vecteur des coordonnées de la position du membre associé au filtre f. Enfin, $\mathbb{E}_f$ est le vecteur des coordonnées de l'estimation du filtre f. Il s'agit donc d'une distance euclidienne entre deux points de l'espace *3D*.

La Jacobienne La Jacobienne de la fonction E_{Part} est aisé à calculer car il s'agit de calculer la variation de $\boldsymbol{O}_f$ par rapport aux paramètres articulaires. Ce n'est rien d'autre que le mouvement d'un point rigidement attaché à un objet articulé et que nous avons abordé dans le chapitre 3 avec l'équation (3.85).

8.3.2 Perspectives

Le filtrage particulaire, tel que nous l'avons présenté, souffre de deux faiblesses :

- Absence d'a priori pour le modèle de mouvement. Pour que l'estimation du mouvement soit correcte, il faudrait intégrer au modèle de mouvement une connaissance a priori (une mesure du mouvement). Nous pourrions estimer le Flot Optique de l'acteur dans les images. Cependant, l'acteur est très peu texturé et la résolution des mains et des pieds est très faible dans les images. L'estimation du flot peut donc s'avérer difficile. Nous avons testé une implémentation pyramidale de l'estimation du flot avec l'algorithme proposé dans [21], actuellement sans grand succès.
- Le référentiel dans lequel nous estimons le mouvement peut être modifié. En effet, nous effectuons la modélisation du mouvement des membres dans le référentiel du monde. Nous pourrions effectuer la modélisation dans un référentiel associé au corps (comme par exemple le repère du pelvis). Pour l'exemple de la marche et plus précisément des pieds, le modèle de mouvement deviendrait alors sinusoïdal et donc plus discriminant que ce que nous traitons actuellement.
- Enfin, actuellement, nous n'avons pas introduit de contraintes sur les filtres. En considérant le changement de référentiel précédent, nous pouvons introduire des contraintes sur l'exploration des particules comme une distance maximum entre les filtres (les mains ne peuvent être distante de plus d'une certaines valeur par exemple). Enfin, nous pouvons introduire des contraintes de non collision entre les filtres. Cette dernière contrainte pourrait empêcher des effets comme ceux observés avec les pieds (attache de deux filtres au même pied).

Chapitre 9

Conclusion

Nous avons atteint dans cette thèse notre objectif de mettre en oeuvre un système de capture de mouvement pour l'animation *3D* qui ne nécessite aucune sorte de marqueurs. A la lumière de ce travail, nous pouvons proposer un certain nombre de perspectives.

9.1 Perspectives à court terme

Dans cette partie, nous proposons des directions de travail, non pas novatrices mais plutôt permettant d'achever certains travaux démarrés pendant la thèse.

Distance de Hausdorff Parmi les contributions présentées, nous avons montré que nous pouvions utiliser la distance de Hausdorff en gardant l'aspect symétrique de celle-ci (la distance du modèle à l'image et de l'image au modèle) (c.f. chapitre 5 section 5.2.2.1). Pendant cette thèse, nous avons implémenté les outils nécessaires pour calculer les deux termes de la somme. Cependant, les résultats proposés n'utilisent que le terme de la distance du modèle à l'image. Concernant le second terme, nous n'avons pas intégré le calcul de l'erreur et de sa Jacobienne dans la fonction de coût globale minimisée. L'intégration de ce second terme dans la fonction de minimisation globale permettra de rendre la minimisation plus robuste aux minima locaux. Les premiers essais démarrés tardivement sont prometteurs.

Suivi spécifique des pieds, des mains et de la tête Dans le chapitre 8, nous avons proposé une approche pour améliorer le suivi des pieds, des mains et de la tête. Nous avons mis en place un filtrage particulaire que nous avons testé en dehors du processus d'estimation de la pose. Les résultats proposés dans le chapitre 8 sont donc indépendants de l'estimation de la pose. Nous pensons qu'intégrer le filtrage particulaire dans le processus global d'estimation de la pose permettrait de stabiliser le suivi du mouvement.

Les limites articulaires Nous avons vu qu'au cours de la minimisation, nous n'imposions pas de limites articulaires. Les travaux de l'UHB permettent de corriger certaines erreurs qui apparaissent et notamment les rotations autour d'axes de symétrie des primitives géométriques. Cependant, dans certains cas, les violations de limites articulaires ne peuvent pas être détectées et donc corrigées. Comme nous l'avons dit, les violations de contrainte importantes sont le signe d'un échec du suivi. Cependant, dans certains cas, les violations sont faibles, n'induisent pas en erreur l'ensemble du suivi mais rendent ce dernier inutilisable pour de l'animation. Nous pouvons prendre l'exemple d'une jambe tendue, où le genoux serait déplié de plus de 180°. Corriger ces violations rendraient le mouvement plus réaliste. Des travaux, comme ceux proposés dans [70] sont des directions intéressantes pour permettre de contraindre le mouvement articulaire lors de l'estimation des paramètres articulaires.

Les évaluations Dans le chapitre 6, nous proposons une comparaison qualitative (visuelle) de nos résultats avec ceux obtenus avec un système à marqueurs (VICON). Il manque une comparaison quantitative des résultats. Nous avons vu que plusieurs difficultés doivent être résolues pour effectuer cette comparaison. Cela peut faire l'objet d'un futur travail dans le cadre d'un stage puisque des thèmes comme la vision, le traitement du signal uni-dimensionnel (pour la synchronisation) peuvent être abordés.

Robustesse pour l'industrie L'objectif du projet SEMOCAP est de proposer un système pouvant servir dans l'industrie du jeu. Nous avons proposé un système avec plusieurs contributions scientifiques. Pour rendre le processus beaucoup plus robuste, plusieurs éléments seraient à rajouter. Le premier élément est probablement la prise en compte de marqueurs pour rendre plus robuste l'estimation du mouvement. Les marqueurs peuvent être pris en compte dans le processus d'estimation au même titre que les contours. En effet, il s'agit de suivre des points rigidement attachés à l'acteur. Nous avons donné au cours de cette thèse l'ensemble des éléments pour introduire ces marqueurs dans le processus d'estimation. Le second point à améliorer est le vêtement pour effectuer la capture du mouvement. Dans l'ensemble des exemples que nous donnons, le personnage est habillé d'un costume rouge uniforme. Ce costume n'est pas idéal pour effectuer la capture du mouvement. Un costume juste au corps avec des couleurs distinctes pour chacun des membres permettrait aussi d'améliorer le suivi. D'une part, le modèle *3D* correspondra mieux et plus facilement à la morphologie de l'acteur et d'autre part nous pourrions introduire un modèle de couleur pour l'estimation du mouvement. L'utilisation du filtrage particulaire pour le suivi des pieds, des mains et de la tête pourrait alors être étendu au cas de l'ensemble des membres.

9.2 Perspectives à moyen terme

Dans la section précédente, nous avons proposé des perspectives immédiates à ce travail qui permettraient de finaliser les travaux en cours. Nous allons maintenant voir des extensions à plus long terme pour compléter ce travail.

Le suivi de points d'intérêts Nous avons proposé dans cette thèse une approche utilisant les contours apparents pour effectuer le suivi du mouvement multi-caméras. Nous avons évoqué les problèmes que nous pouvons avoir à détecter les contours dans certaines situations (vêtements trop texturés par exemple). Nous pouvons envisager d'employer le même principe d'estimation des paramètres mais en utilisant des points d'intérêts détectés dans les images simultanément aux contours. Nous avons posé l'ensemble des bases mathématiques nécessaire pour effectuer le suivi à l'aide de points d'intérêts. Cependant, une mise en oeuvre effective suppose un travail pour effectuer le suivi des points et l'estimation de leurs coordonnées locales dans le modèle *3D*.

Les contraintes temporelles Nous avons proposé une méthode de suivi du mouvement linéaire dans le sens où nous effectuons le suivi du mouvement avec une ligne de temps uni-directionnelle. Effectuer le suivi en plusieurs passes, ou en intégrant sur plusieurs images permettrait probablement d'améliorer la stabilité des résultats. Plusieurs points de vue peuvent être adoptés :

- Le suivi en plusieurs passes. Beaucoup de travaux proposent d'effectuer le suivi avec plusieurs niveaux de détail dans le squelette permettant d'affiner l'estimation à chaque passe.
- Intégration temporelle du suivi. Nous pouvons envisager d'effectuer le suivi à l'aide de plusieurs images en même temps. Nous pouvons par exemple prendre deux images clefs et interpoler le mouvement entre les images en s'aidant des observations ou encore effectuer un *bundle adjustement* de l'ensemble des paramètres de pose sur quelques images. Cette intégration temporelle permettrait de lisser le suivi du mouvement afin d'éliminer les effets d'oscillation du modèle.

Par exemple, nous avons proposé un suivi spécifique pour les pieds, les mains et la tête. Si nous effectuons le suivi sur toute la séquence vidéo, nous obtenons les trajectoires en *3D* de ces membres. Nous pouvons alors utiliser ces trajectoires pour aider à estimer l'ensemble des paramètres du modèle. Connaissant la position de certains membres au cours du temps, comment déterminer l'ensemble des paramètres de pose du modèle ? Nous pouvons penser à la cinématique inverse, mais aussi à des méthodes de contrôle de trajectoire pouvant s'appuyer sur les travaux proposés dans cette thèse.

Les variables de contrôle Pour effectuer le suivi du mouvement humain, nous proposons d'utiliser l'estimation de **quarante quatre** paramètres. Ne peut-on pas réduire le nombre de paramètres ? Les travaux de bio-mécanique tendent à montrer que certains degrés de liberté sont en réalité très corrélés entre eux. Nous pouvons donc utiliser des espaces de dimension réduite pour effectuer le suivi. La difficulté majeure consiste alors à estimer les paramètres de cet espace qui ne sont pas toujours observables.

Intégration de la *3D* pour l'estimation Des travaux récents de notre groupe pourraient être utilisés afin d'améliorer le suivi du mouvement. Les travaux proposés dans [113] abordent la capture du mouvement à l'aide d'une reconstruction par *patch 3D* de l'acteur. L'intégration de données *3D* à des données *2D* permettrait de rendre

robuste le suivi du mouvement. Une approche alliant la *3D* et la *2D* est proposée dans [84] et montre l'aspect prometteur de cette alliance.

Initialisation L'initialisation du suivi dans des séquences vidéo est un problème rencontré dans de nombreuses applications. De cette étape dépend la précision et le bon déroulement du suivi.

Pendant cette thèse, nous avons proposé une méthode d'initialisation de la pose pour la première image de la séquence vidéo. Cette initialisation fait l'hypothèse que l'acteur adopte une pose prédéfinie. Cette initialisation contraint donc le mouvement initial de l'acteur ce qui peut ne pas être acceptable dans certains cas comme lors du suivi du mouvement sportif. Une initialisation plus souple permettrait donc de laisser l'acteur plus libre de ses mouvements mais permettrait aussi d'utiliser des séquences vidéo de bases de données externes (une initialisation manuelle est toujours possible). Des travaux comme ceux proposés dans [125] ou encore plus récemment [121] permettraient d'effectuer, moyennant une adaptation à la *3D*, une détection de la pose initiale de l'acteur.

Intégration de marqueurs magnétiques Pendant la dernière année de thèse, nous avons démarré un partenariat avec l'équipe BiPop et le cea pour utiliser des centrales inertielles pour le suivi du mouvement. Nous avons proposé et encadré un stage sur l'intégration des centrales pour le suivi. Les premiers résultats obtenus sont encourageants. L'intégration de ces centrales dans le processus d'estimation du mouvement permettrait de contraindre l'estimation du mouvement et surtout d'aider à ré-initialiser correctement le suivi du mouvement lors d'éventuels échecs de l'estimation des paramètres de pose.

Troisième partie

Annexes

Annexe A

Annexes du chapitre 3

A.1 La matrice exponentielle

Nous allons montrer que $e^{[\boldsymbol{\omega}]_\times \theta}$ est bien une matrice de rotation. Puisqu'il s'agit d'une matrice exponentielle, nous pouvons la développer sous la forme d'un développement en série :

$$e^{[\boldsymbol{\omega}]_\times \theta} = \mathbf{I} + [\boldsymbol{\omega}]_\times \theta + \frac{([\boldsymbol{\omega}]_\times \theta)^2}{2} + \ldots + \frac{([\boldsymbol{\omega}]_\times \theta)^n}{n!} + \ldots \tag{A.1}$$

Cependant, en pratique, ce développement n'est pas exploitable. Nous allons donc simplifier l'expression (A.1) en considérant les deux identités suivantes :

$\forall \boldsymbol{a} \in \mathbb{R}^3$:

$$[\boldsymbol{a}]_\times^2 = \boldsymbol{a}\boldsymbol{a}^\top - \|\boldsymbol{a}\|^2 \mathbf{I} \tag{A.2}$$

$$[\boldsymbol{a}]_\times^3 = -\|\boldsymbol{a}\|^2 \mathbf{I} \tag{A.3}$$

Ces deux identités nous permettent de calculer, par récursivité, $[\boldsymbol{a}]_\times^n$ pour tout $n \in \mathbb{N}$. Par exemple, pour $n = 7$:

$$[\boldsymbol{a}]_\times^7 = [\boldsymbol{a}]_\times^3 [\boldsymbol{a}]_\times^3 [\boldsymbol{a}]_\times \tag{A.4}$$

$$= \|[\boldsymbol{a}]_\times\|^4 [\boldsymbol{a}]_\times \tag{A.5}$$

En considérant ces identités, nous pouvons réécrire l'équation (A.1) sous la forme :

$$e^{[\boldsymbol{\omega}]_\times \theta} = \mathbf{I} + (\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \ldots)[\boldsymbol{w}]_\times + (\frac{\theta^2}{2!} - \frac{\theta^4}{4!} + \ldots)[\boldsymbol{w}]_\times^2 \tag{A.6}$$

D'autre part, les développements limités des fonctions cosinus et sinus sont de la forme :

$$\cos(\theta) = 1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} + \ldots$$

$$\sin(\theta) = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} + \ldots .$$

Nous pouvons donc ré-écrire (A.1) sous la forme :

$$\boxed{e^{[\boldsymbol{\omega}]_\times \theta} = \mathbf{I} + \sin\theta[\boldsymbol{\omega}]_\times + (1-\cos\theta)[\boldsymbol{\omega}]_\times^2.} \tag{A.7}$$

L'équation (A.7) est connue sous le nom de formule de Rodriguez qui est l'expression d'une matrice de rotation.

A.2 Les conventions d'Euler et de Cardan

Il existe 3 conventions d'Euler que nous allons donner maintenant. Pour les besoins de la description, notons $(O_1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ le repère associé à $\mathcal{F}$ et $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ le repère associé à $\mathcal{M}$.

Convention angle fixe : Soit (D) la droite décrivant l'intersection des plans décrits par $(O_1, \boldsymbol{i}, \boldsymbol{j})$ et $(O_2, \boldsymbol{l}, \boldsymbol{m})$ (c.f. figure A.1). Alors,
- Θ est l'angle entre l'axe orienté par $\boldsymbol{i}$ et la droite (D)
- Φ est l'angle entre l'axe orienté par $\boldsymbol{k}$ et l'axe orienté par $\boldsymbol{n}$
- Ψ est l'angle entre la droite (D) et l'axe orienté par $\boldsymbol{l}$.

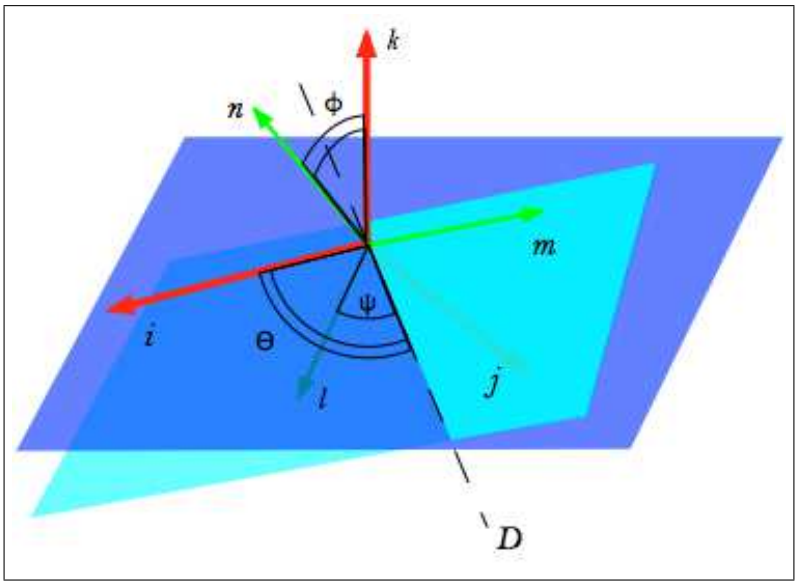


FIG. A.1: Illustration du formalisme d'Euler pour le cas fixe.

Convention avec axes de rotation fixes : Si nous considérons $\mathcal{F}$ et $\mathcal{M}$ confondus dans un premier temps (c.f. figure A.2-a), la rotation avec axes fixes se fait de la manière suivante :
- rotation de $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ autour de $\boldsymbol{k}$ d'angle Θ (c.f. figure A.2-b)
- rotation de $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ autour de $\boldsymbol{i}$ d'angle Φ (c.f. figure A.2-c)
- rotation de $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ autour de $\boldsymbol{k}$ d'angle Ψ (c.f. figure A.2-d)

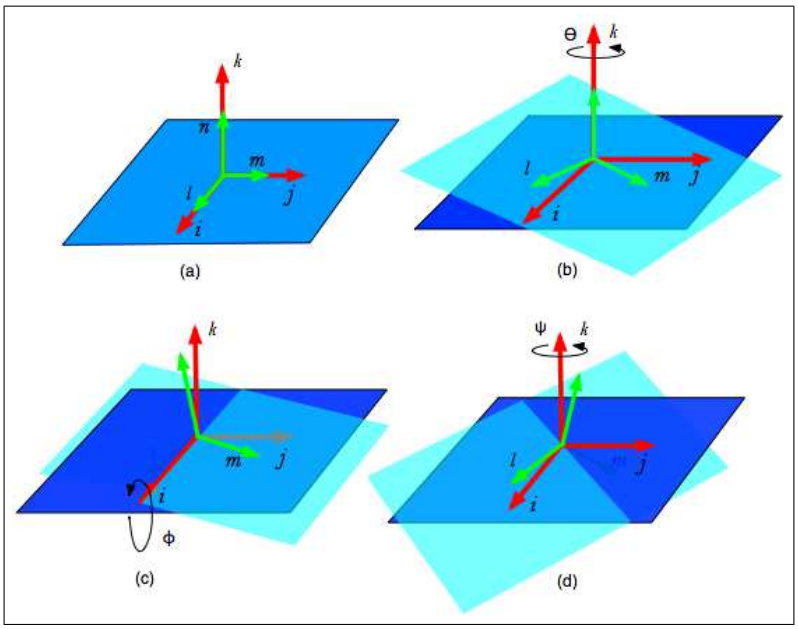


FIG. A.2: Illustration du formalisme d'Euler pour le cas où les axes de rotation sont fixes.

Convention avec axes de rotation en mouvement : Si nous considérons les deux repères confondus dans un premier temps (c.f. figure A.3-a), la rotation avec les axes de rotation en mouvement se fait de la manière suivante :
- rotation de $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ autour de $\boldsymbol{n}$ d'angle Ψ (c.f. figure A.3-b)
- rotation de $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ autour de $\boldsymbol{l}$ (qui a subi une rotation) d'angle Φ (c.f. figure A.3-c)
- rotation de $(O_2, \boldsymbol{l}, \boldsymbol{m}, \boldsymbol{n})$ autour de $\boldsymbol{n}$ (qui a subi deux rotations) d'angle Θ (c.f. figure A.3-d)

Dans la littérature, afin de distinguer les conventions, les axes sont nommés en majuscules pour les rotations d'axes fixes et en minuscules pour les rotations d'axes en mouvement. D'autre part, les rotations sont nommées en fonction de l'ordre des rotations. Par exemple, la dénomination $\mathbf{R}_{KJK}$ représente une première rotation autour de l'axe K, puis autour de l'axe J puis autour de l'axe K en convention axes fixes. Tandis que $\mathbf{R}_{iji}$ représente une première rotation autour de l'axe i puis autour de j et enfin autour de i en convention axes mobiles.

Il existe d'autres conventions d'utilisation des angles d'Euler utilisées notamment en navigation. Contrairement aux conventions précédentes qui sont de type KJK, les conventions de navigation sont de type roulis, tangage, lacet (roll, pitch, yaw) soit kji. Ce type de convention à l'avantage d'être plus intuitif que la convention d'Euler. Il est connu sous le nom d'angle de Cardan.

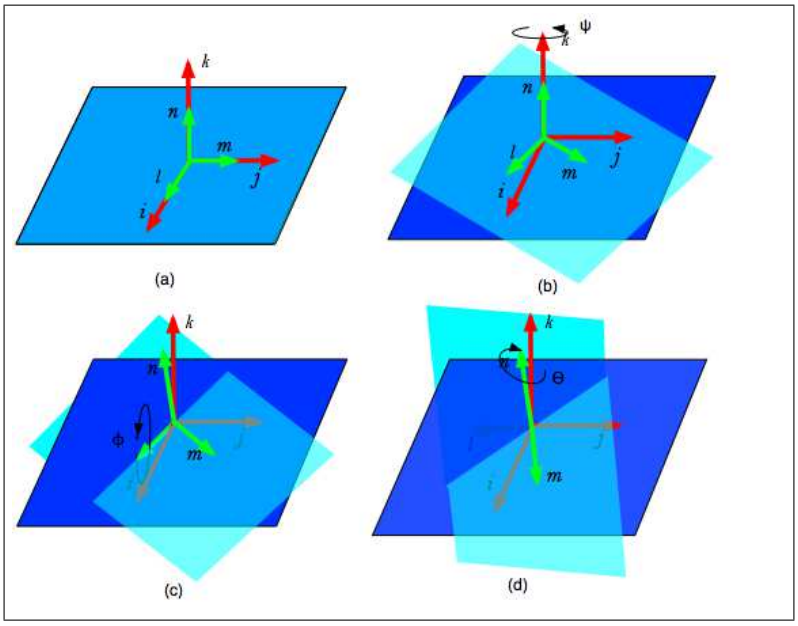


FIG. A.3: Illustration du formalisme d'Euler pour le cas où les axes de rotation sont mobiles.

La matrice de rotation déduite des angles d'Euler (ou de Cardan) se construit en composant les rotations autour de chaque axe. Par exemple :

$$\mathbf{R}_{KJK}(\Theta,\Phi,\Psi) = \mathbf{R}_K(\Psi)\mathbf{R}_J(\Phi)\mathbf{R}_K(\Theta), \quad (A.8)$$

$$\mathbf{R}_{kji}(\Theta,\Phi,\Psi) = \mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)\mathbf{R}_k(\Theta), \quad (A.9)$$

où $R_A(\alpha)$ est la rotation autour de l'axe A et d'angle α.

A.3 Calcul des angles d'Euler

Dans le cadre de nos travaux, nous sommes contraints de souvent effectuer des transformations successives pour déterminer les positions et orientations de contours ou de parties du modèle *3D*. Cependant, il est plus simple d'effectuer les estimations à partir des positions et des angles d'Euler. Nous avons dû mettre en place des routines de conversion permettant d'extraire à partir de matrices de transformation données les angles d'Euler et le vecteur de positions. Pour le second terme, il s'agit, en général, de la dernière colonne de la matrice de transformation. Pour les angles d'Euler, il faut les extraire de la matrice de rotation.

Nous avons vu dans le chapitre 3 qu'il existait plusieurs conventions d'Euler ou encore les angles de Cardan. Nous allons ici nous restreindre à l'extraction des angles pour la convention kji (qui correspond à la convention d'axe en mouvement avec la rotation autour de l'axe $\boldsymbol{k}$ puis $\boldsymbol{j}$ puis $\boldsymbol{i}$). Cependant, nous utilisons un code générique

écrit par Ken Shoemake en 1993 et pris de Graphics Gems 4. Ce code permet d'extraire les angles d'une matrice de rotation avec n'importe quelle convention.

Dans un premier temps, nous calculons la matrice $\mathbf{R}_{kji}$. Puis nous donnons les formules pour les angles d'Euler associés. Enfin, nous montrons que pour extraire les angles pour la convention d'Euler axes fixes, il suffit de les extraire pour la convention axe en mouvement.

A.3.1 La matrice de rotation kji

La matrice $\mathbf{R}_{kji}$ s'écrit :

$$\mathbf{R}_{kji} = \mathbf{R}_i(\Psi)\mathbf{R}_j(\Phi)\mathbf{R}_k(\Theta), \tag{A.10}$$

$$\mathbf{R}_i(\Psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\Psi) & -s(\Psi) \\ 0 & s(\Psi) & c(\Psi) \end{bmatrix}, \quad \mathbf{R}_j(\Phi) = \begin{bmatrix} c(\Phi) & 0 & s(\Phi) \\ 0 & 1 & 0 \\ -s(\Phi) & 0 & c(\Phi) \end{bmatrix} \text{ et}$$

avec

$$\mathbf{R}_k(\Theta) = \begin{bmatrix} c(\Theta) & -s(\Theta) & 0 \\ s(\Theta) & c(\Theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{où } c = \cos \text{ et } s = \sin.$$

En développant et simplifiant, nous obtenons :

$$\mathbf{R}_{kji} = \begin{bmatrix} c(\Phi)c(\Theta) & -s(\Theta)c(\Phi) & s(\Phi) \\ c(\Psi)s(\Theta) + s(\Psi)s(\Phi)c(\Theta) & c(\Psi)c(\Theta) - s(\Psi)s(\Phi)s(\Theta) & -c(\Phi)s(\Psi) \\ s(\Psi)s(\Theta) - c(\Psi)s(\Phi)c(\Theta) & s(\Psi)c(\Theta) + c(\Psi)s(\Theta)s(\Phi) & c(\Psi)c(\Phi) \end{bmatrix} \tag{A.11}$$

$$= \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{A.12}$$

A.3.2 Les angles d'Euler

$\mathbf{R}_{kji}$ reste inchangée si nous effectuons les transformations suivantes :

$$\begin{aligned} \Theta &\rightarrow \pi + \Theta \\ \Phi &\rightarrow \pi - \Phi \\ \Psi &\rightarrow \pi + \Psi \end{aligned}$$

Nous pouvons donc imposer $\Phi \in [-\pi/2; \pi/2]$ et déduire les angles de la matrice :

$$\Theta = \arctan\left(\frac{-\mathbf{R}_{12}}{\mathbf{R}_{11}}\right) \tag{A.13}$$

$$\Phi = \arcsin(\mathbf{R}_{13}) \tag{A.14}$$

$$\Psi = \arctan\left(\frac{-\mathbf{R}_{23}}{\mathbf{R}_{33}}\right) \tag{A.15}$$

A.3.3 Equivalence des conventions

Pour calculer les valeurs des angles pour une rotation avec la convention axe fixe, il suffit de la calculer pour la convention d'axe en mouvement puis d'inverser l'ordre des angles.

Preuve :

Soit $\mathbf{R}_I$, $\mathbf{R}_J$ et $\mathbf{R}_K$ les rotations autour des axes $\boldsymbol{I}$, $\boldsymbol{J}$ et $\boldsymbol{K}$ pour la convention d'axes fixes. Soit $\mathbf{R}_i$, $\mathbf{R}_j$ et $\mathbf{R}_k$ les rotations autour des axes $\boldsymbol{i}$, $\boldsymbol{j}$ et $\boldsymbol{k}$ pour la convention d'axes en mouvement.

Par construction des rotations, nous avons :

$$\begin{aligned}\mathbf{R}_k &= \mathbf{R}_K \\ \mathbf{R}_j &= \mathbf{R}_K\mathbf{R}_J\mathbf{R}_K^{-1} \\ \mathbf{R}_i &= \mathbf{R}_j\mathbf{R}_K\mathbf{R}_I\mathbf{R}_K^{-1}\mathbf{R}_j^{-1} \\ &= \mathbf{R}_K\mathbf{R}_J\mathbf{R}_K^{-1}\mathbf{R}_K\mathbf{R}_I\mathbf{R}_K^{-1}\mathbf{R}_K\mathbf{R}_J^{-1}\mathbf{R}_K^{-1}\end{aligned}$$

Nous avons donc :

$$\begin{aligned}\mathbf{R}_i\mathbf{R}_j\mathbf{R}_k &= \mathbf{R}_K\mathbf{R}_J\underbrace{\mathbf{R}_K^{-1}\mathbf{R}_K}_{\mathbf{I}}\mathbf{R}_I\underbrace{\underbrace{\mathbf{R}_K^{-1}\mathbf{R}_K}_{\mathbf{I}}\mathbf{R}_J^{-1}\underbrace{\mathbf{R}_K\mathbf{R}_K^{-1}}_{\mathbf{I}}\mathbf{R}_J}_{\mathbf{I}}\underbrace{\mathbf{R}_K^{-1}\mathbf{R}_K}_{\mathbf{I}} \\ &= \mathbf{R}_K\mathbf{R}_J\mathbf{R}_I\square \qquad (A.16)\end{aligned}$$

A.4 Jacobien d'un quaternion

La dérivation des quaternions permet d'exprimer la vitesse de rotation d'un objet. Nous considérons alors deux référentiels $\mathcal{O}_1$ et $\mathcal{O}_2$ dont les origines sont confondues.

Soit $\boldsymbol{q}(t) = (q_0(t), q_1(t), q_2(t), q_3(t))$ le quaternion représentant la rotation de $\mathcal{O}_2$ par rapport à $\mathcal{O}_1$. Soit $\dot{\boldsymbol{\Lambda}} = (0, \dot{\lambda}_i, \dot{\lambda}_j, \dot{\lambda}_k)$ le quaternion représentant la vitesse angulaire.

Soit un point X rigidement attaché à $\mathcal{O}_2$. Soit $\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}$ les coordonnées de X dans le repère $\mathcal{O}_1$ exprimées sous forme d'un quaternion ($\widetilde{\boldsymbol{X}}^{\mathcal{O}_1} = (0, \boldsymbol{X}^{\top\mathcal{O}_1})$).

Le déplacement du point X de la position initiale à la position courante s'écrit alors :

$$\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t) = \boldsymbol{q}(t)\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(0)\boldsymbol{q}(t)^{-1}. \qquad (A.17)$$

Si nous différencions (A.17), nous obtenons :

$$\dot{\widetilde{\boldsymbol{X}}}^{\mathcal{O}_1}(t) = \dot{\boldsymbol{q}}(t)\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(0)\boldsymbol{q}^{-1}(t) + \boldsymbol{q}(t)\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(0)\dot{\boldsymbol{q}}^{-1}(t). \qquad (A.18)$$

En combinant (A.17) avec l'équation ci-dessus, nous avons :

$$\dot{\widetilde{\boldsymbol{X}}}^{\mathcal{O}_1}(t) = \dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t)\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t) + \widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t)\boldsymbol{q}(t)\dot{\boldsymbol{q}}^{-1}(t). \tag{A.19}$$

Nous allons maintenant simplifier cette expression.

Nous avons $\|\boldsymbol{q}\|^2 = 1$, donc $\boldsymbol{q}(t)\boldsymbol{q}^{-1}(t) = 1$ et par conséquence :

$$\dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t) + \boldsymbol{q}(t)\dot{\boldsymbol{q}}^{-1}(t) = 0. \tag{A.20}$$

En substituant dans l'équation (A.19), nous avons :

$$\dot{\widetilde{\boldsymbol{X}}}^{\mathcal{O}_1}(t) = \dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t)\widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t) - \widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t)\dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t) \tag{A.21}$$

De plus, la partie réelle de $\dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t)$ est nulle ($\|\boldsymbol{q}\|^2 = 1$) donc la dérivée de la norme est nulle. Or la dérivée est une somme sur la base des imaginaires purs, donc chacun des termes de la somme doit être nulle.

$\dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t)$ est donc un vecteur. Avec les propriété multiplicative des quaternions, l'équation (A.21) peut donc être écrite :

$$\dot{\widetilde{\boldsymbol{X}}}^{\mathcal{O}_1}(t) = 2\dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t) \times \widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t). \tag{A.22}$$

D'autre part, nous avons pu voir dans le chapitre 3 (équation (3.14)) que la vitesse de rotation d'un point pouvait être mise sous la forme :

$$\dot{\widetilde{\boldsymbol{X}}}^{\mathcal{O}_1}(t) = \dot{\boldsymbol{\Lambda}} \times \widetilde{\boldsymbol{X}}^{\mathcal{O}_1}(t). \tag{A.23}$$

Par identification, nous avons donc :

$$\dot{\boldsymbol{\Lambda}} = 2\dot{\boldsymbol{q}}(t)\boldsymbol{q}^{-1}(t) \tag{A.24}$$

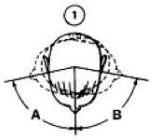
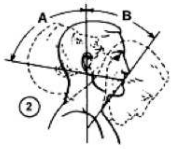
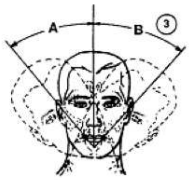
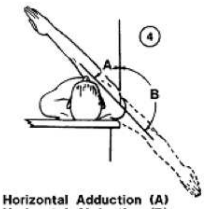
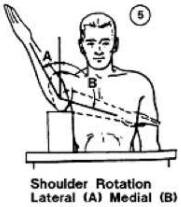
Enfin :

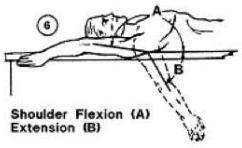
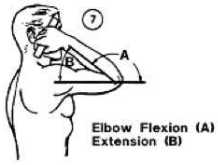
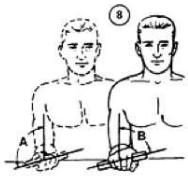
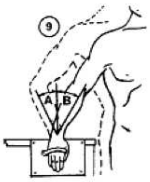
$$\boxed{\dot{\boldsymbol{q}}(t) = 1/2\dot{\boldsymbol{\Lambda}}\boldsymbol{q}(t)} \tag{A.25}$$

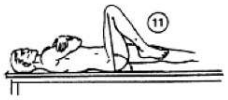
Annexe B

Annexes du chapitre 4

B.1 Limitations Articulaires

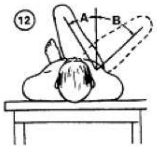
Figure	Joint movement (note b)	Range of motion (degrees)			
		Males (note a)		Female (note a)	
		5th percentile	95th percentile	5th percentile	95th percentile
1	Neck, rotation right (A)	73.3	99.6	74.9	108.8
 Neck Rotation Right (A) Left (B)	Neck, rotation left (B)	74.3	99.1	72.2	109.0
2	Neck, flexion (B)	34.5	71.0	46.0	84.4
 Neck Extension [A] Flexion (B)	Neck, extension (A)	65.4	103.0	4.9	103.0
3	Neck, lateral bend right (A)	34.9	63.5	37.0	63.2
 Neck Lateral Bend Right (A) Left (B)	Neck, lateral bend left (B)	35.5	63.5	29.1	77.2
4	Shoulder, abduction	173.2	188.7	172.6	192.9
 Horizontal Adduction (A) Horizontal Abduction (B)					
5	Shoulder, rotation lateral (A)	46.3	96.7	53.8	85.8
 Shoulder Rotation Lateral (A) Medial (B)	Shoulder, rotation medial (B)	90.5	126.6	95.8	130.9
6	Shoulder, flexion	164.4	210.9	152.0	217.0

	(A)				
 Shoulder Flexion (A) Extension (B)	Shoulder, extension (B)	39.6	83.3	33.7	87.9
7	Elbow, flexion (A)	140.5	159.0	144.9	165.9
 Elbow Flexion (A) Extension (B)					
8	Forearm, pronation (B)	78.2	116.1	82.3	118.9
 Forearm Supination (A) Pronation (B)	Forearm, supination (A)	83.4	125.8	90.4	139.5
9	Wrist, radial bend (B)	16.9	36.7	16.1	36.1
 Wrist Ulnar Bend (A) Radial Bend (B)	Wrist, ulnar bend (A)	18.6	47.9	21.5	43.0
10	Wrist, flexion (A)	61.5	94.8	68.3	98.1
 Wrist Flexion (A) Extension (B)	Wrist, extension (B)	40.1	78.0	42.3	74.7
11	Hip, flexion	116.5	148.0	118.5	145.0



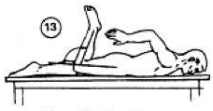
Hip Flexion

12	Hip, abduction (B)	26.8	53.5	27.2	55.9



Hip Adduction (A)
Abduction (B)

13	Knee, flexion	118.4	145.6	125.2	145.2



Knee Flexion, Prone

14	Ankle, plantar extension (A)	36.1	79.6	44.2	91.1
	Ankle, dorsi flexion (B)	8.1	19.9	6.9	17.4

Ankle Plantar Extension (A)
Dorsi Flexion

Notes:

a. Data was taken 1979 and 1980 at NASA-JSC by Dr. William Thornton and John Jackson. The study was made using 192 males (mean age 33) 22 females (mean age 30) astronaut candidates (see Reference 365).

b. Limb range is average of right and left limb movement.

B.2 Paramétrage et projection des ellipsoïdes

Nous pouvons améliorer la modélisation du corps humain en remplaçant les cônes modélisant la tête et les mains par des ellipsoïdes. Nous allons aborder ici le paramétrage des ellipsoïdes ainsi que le paramétrage de leurs contours extrémaux dans les images.

B.2.1 Le paramétrage

Définition Une quadrique Q est une surface implicite d'ordre 2 dans l'espace *3D*. Elle peut être représentée en coordonnées homogènes par une matrice de dimension 4×4 symétrique $\mathbf{Q}$. Tout point de la surface X de coordonnées homogènes $\overline{\boldsymbol{X}}$ satisfait l'équation suivante :

$$\overline{\boldsymbol{X}}^{\top} \mathbf{Q} \overline{\boldsymbol{X}} = 0 \tag{B.1}$$

Parmi les surfaces quadriques, nous pouvons citer les ellipsoïdes, les paraboloïdes elliptiques, les cylindres hyperboliques...

Les ellipsoïdes sont une sous-classe de quadriques dont la matrice $\mathbf{Q}$ est de la forme :

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{\alpha^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\beta^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\gamma^2} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{B.2}$$

La forme implicite des ellipsoïdes est :

$$(\frac{x}{\alpha})^2 + (\frac{y}{\beta})^2 + (\frac{z}{\gamma})^2 = 1, \tag{B.3}$$

Si $\alpha = \beta = \gamma$, alors nous avons un sphéroïde dont le rayon est $\sqrt{\alpha}$.

Pour information, l'équation paramétrique d'un ellipsoïde est de la forme :

$$\boldsymbol{X}(\theta, \phi) = \begin{bmatrix} \alpha \cos(\theta) \sin(\phi) \\ \beta \sin(\theta) \sin(\phi) \\ \gamma \cos(\phi) \end{bmatrix} \tag{B.4}$$

avec $\theta \in [0 \dots 2\pi]$ et $\phi \in [0 \dots \pi]$.

B.2.2 Projection des ellipsoïdes

Nous allons aborder dans cette annexe la projection de l'ellipsoïde sur le plan image. La projection de l'ellipsoïde dans le plan image est une ellipse. Nous allons d'une part le montrer et d'autre part déterminer les paramètres de cette ellipse en fonction de la pose et des paramètres de calibrage de la caméra.

De manière intuitive, il s'agit dans un premier temps de déterminer le lieu des points ou le rayon de vue est tangent à la surface de l'ellipsoïde. Pour cela, nous allons utiliser la géométrie projective. Cette formulation du problème est plus simple que celle utilisant l'équation explicite de la contrainte de tangence telle que proposée dans le chapitre 4 au paragraphe 4.3.2. En effet, les simplifications intervenant dans le cas de la projection du cône n'apparaissent pas dans le cas des ellipsoïdes.

Posons $\mathbf{Q}$ la matrice caractéristique et canonique d'un ellipsoïde. Alors tous points X de l'ellipsoïde, dont les coordonnées sont exprimées dans le repère canonique de l'ellipsoïde vérifie l'équation :

$$\overline{\boldsymbol{X}}^{\top}\mathbf{Q}\overline{\boldsymbol{X}} = 0. \tag{B.5}$$

Nous avons vu que la matrice $\mathbf{Q}$ est une matrice symétrique. Nous pouvons écrire celle-ci sous la forme :

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \boldsymbol{b} \\ \boldsymbol{b}^{\top} & c \end{bmatrix} \tag{B.6}$$

Nous nous plaçons dans le repère de la caméra. On note $\mathbf{R}_e$ et t_e l'orientation et la position de l'ellipsoïde dans le repère de la caméra. On note alors $\mathbf{M}_e$ la configuration de l'ellipsoïde. Alors l'ellipsoïde a pour matrice caractéristique dans le repère de la caméra :

$$\widetilde{\mathbf{Q}} = \mathbf{M}_e^{-\top}\mathbf{Q}\mathbf{M}_e^{-1}. \tag{B.7}$$

Nous allons définir la condition pour laquelle un point de l'image appartient à la projection de l'ellipsoïde. Soit x_c un point de l'image dont les coordonnées homogènes sont $\overline{\boldsymbol{x}}_c$. Le centre optique et ce point définissent un rayon de vue $\overline{\boldsymbol{V}}(d) = (\overline{\boldsymbol{x}}_c, d)^{\top}$, où d est l'inverse de la profondeur du point. Pour déterminer le contour occultant, il suffit de résoudre l'équation suivante :

$$\overline{\boldsymbol{V}}^{\top}\widetilde{\mathbf{Q}}\overline{\boldsymbol{V}} = 0, \tag{B.8}$$

En développant cette équation, nous obtenons une équation du second degré en d :

$$\widetilde{c}d^2 + 2\widetilde{\boldsymbol{b}}^{\top}\boldsymbol{x}_c d + \boldsymbol{x}_c^{\top}\widetilde{\mathbf{A}}\boldsymbol{x}_c = 0. \tag{B.9}$$

Cette équation, pour un point $\overline{\boldsymbol{x}}$ donné, a une solution unique si son discriminant est nul :

$$\overline{\boldsymbol{x}}_c^{\top}(\widetilde{\boldsymbol{b}}\widetilde{\boldsymbol{b}}^{\top} - \widetilde{c}\widetilde{\mathbf{A}})\overline{\boldsymbol{x}}_c = 0. \tag{B.10}$$

Cette condition est vérifiée si $\overline{\boldsymbol{x}}_c$ appartient à la conique définie par : $\mathbf{E} = \widetilde{\boldsymbol{b}}\widetilde{\boldsymbol{b}}^{\top} - \widetilde{c}\widetilde{\mathbf{A}}$. Dans notre cas, $\mathbf{E}$ définit une ellipse et :

$$\widetilde{\mathbf{A}} = \mathbf{R}\mathbf{A}\mathbf{R}^{-1} \tag{B.11}$$

$$\widetilde{\boldsymbol{b}} = \mathbf{R}\boldsymbol{b}\mathbf{R}^{-1} \tag{B.12}$$

$$\widetilde{c} = (\mathbf{R}^{-1}\boldsymbol{t})^{-1}\mathbf{A}(\mathbf{R}^{-1}\boldsymbol{t}) + c. \tag{B.13}$$

Nous avons donc l'équation de l'ellipse dans le plan image. Elle dépend des paramètres extrinsèques de la caméra, de la position et de l'orientation de l'ellipsoïde dans le repère du monde.

Nous pouvons donc étudier la variation de ce contour dans les images en fonction de la variation des paramètres de pose de l'ellipsoïde. Nous ne donnerons pas la forme explicite du Jacobien, cependant, nous donnerons les étapes de calcul.

En développant l'équation (B.10) nous obtenons l'équation homogène de l'ellipse. Cette équation est de la forme :

$$au^2 + bv^2 + ch^2 + 2duv + 2evh + 2fuh, \tag{B.14}$$

en posant $\overline{\boldsymbol{x}} = (u, v, h)^\top$. a, b, c, d, e, f sont des scalaires dont les valeurs dépendent des paramètres de pose de l'ellipsoïde, des paramètres de l'ellipsoïde ainsi que de ceux de la caméra.

Nous pouvons noter que pour $h = 1$, nous avons l'équation affine d'une ellipse. Un changement de repère dans le plan affine $z = 1$ permet de se ramener à l'équation canonique d'une ellipse.

$$(\frac{u}{\alpha})^2 + (\frac{v}{\beta})^2 = 1 \tag{B.15}$$

Nous pouvons alors déduire α et β de l'équation affine de l'ellipse. Nous avons $\alpha^2 = -c/a$ et $\beta^2 = -c/b$.

Nous pouvons alors obtenir l'équation paramétrique de l'ellipse dans l'image (dans le repère canonique) :

$$\boldsymbol{x}(\boldsymbol{\Phi}, \theta) = \begin{pmatrix} \alpha(\boldsymbol{\Phi}) \cos\theta \\ \beta(\boldsymbol{\Phi}) \sin\theta \end{pmatrix}, \tag{B.16}$$

où $\boldsymbol{\Phi}$ est le vecteur des paramètres de pose de l'ellipsoïde dans le repère caméra.

Nous avons donc l'équation paramétrique de l'ellipse. Pour déterminer la variation de l'ellipse en fonction des paramètres de pose de l'ellipsoïde, nous devons calculer la variation du changement de repère nécessaire pour obtenir l'équation canonique de l'ellipse puis calculer la dérivée de l'équation paramétrique.

Annexe C

Formats descriptifs de la chaîne articulaire

Dans le cadre du projet SEMOCAP, nous avons eu à interagir avec l'Université de Haute Bretagne (UHB). Après avoir effectué la capture du mouvement et donc l'estimation des paramètres de l'acteur, l'UHB a permis d'effectuer les traitements nécessaires pour adapter le mouvement à un nouvel acteur. Le problème n'est pas simple (c.f. chapitre 6 section 6.1.3).

Pour pouvoir échanger les données de mouvement entre les partenaires, nous avons utilisé le format d'échange BVH. Dans un premier temps nous allons donc décrire ce format de fichier. Nous verrons alors que celui-ci ne répond pas à tous nos besoins, nous avons donc ajouté un bloc permettant de stocker des informations nécessaires que nous décrirons. De plus, nous avons vu que nous n'utilisions pas les conventions d'orientation préconisées par la norme H-ANIM (c.f. chapitre section 3.5.1) pour des raisons pratiques. Or le logiciel MKM développé par l'UHB utilise cette norme. Nous avons donc mis en place une méthode de conversion d'une convention à l'autre que nous expliciterons dans la dernière partie de cette annexe.

C.1 Normes et Fichiers d'échange

Le format de fichier d'échange peut être dissocié de la norme utilisée. La norme la plus couramment utilisée est la norme H-ANIM. Cette norme propose aussi un format de fichier d'échange au format VRML avec une grammaire associée. H-ANIM donne d'une part les dimensions standards du squelette, les modèles utilisés pour modéliser les différentes parties du corps, les contraintes articulaires et d'autre part définit clairement la position de référence initiale avec la convention d'orientation d'orientation des repères. Cette norme, très spécifique, permet un échange de données facilité.

Le format VRML associé à la norme H-ANIM permet une description exhaustive du modèle utilisé. Des formats de fichier standards plus simples existent. Nous pouvons

citer les formats de type BVH, ASK/SDL (créés par les sociétés Biovision et Alias), les fichiers de type C3D ou encore CSM, *etc*. Nous utilisons le format de type BVH. Ce format de fichier est très permissif pour la représentation du squelette et de son mouvement. Le fichier est partagé en 2 blocs. Le premier bloc contient une description du squelette avec le placement relatif des articulations et les degrés de liberté de chacune des articulations. Le second bloc contient les informations relatives au mouvement. Plusieurs champs apparaissent : le nombre d'images de la séquence du mouvement, la fréquence d'exécution du mouvement et enfin les paramètres articulaires du squelette pour tout le mouvement. Ce format de fichier n'est pas une norme, et la norme H-ANIM peut être utilisée pour la description du squelette.

Cependant, le fichier BVH ne contient pas toute l'information nécessaire pour représenter notre modèle *3D*. Pour des raisons pratiques permettant de faciliter l'échange de données (entre les protagonistes du projet SEMOCAP), Loïc Lefort (ingénieur du projet SEMOCAP) a rajouté dans ce fichier un troisième bloc permettant de décrire les primitives volumétriques, les contraintes de symétries, *etc*. utilisées pour modéliser l'acteur. Ce troisième bloc permet d'échanger les données entre le logiciel de capture de l'acteur (création du modèle *3D*) et le logiciel de capture du mouvement. Nous donnons maintenant un exemple de fichier BVH que nous commentons ensuite.

Exemple de BVH

```
HIERARCHY
ROOT Root
{
  OFFSET 503.306680 255.923672 639.058859
  CHANNELS 6 Xposition Yposition Zposition Xrotation Yrotation Zrotation
  JOINT Sacroiliac
  {
    OFFSET 0.000000 0.000000 180.900000
    CHANNELS 3 Xrotation Yrotation Zrotation
    JOINT Dorsal
    {
      OFFSET 0.000000 0.000000 36.180000
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT l_Sternoclavicular
      {
        OFFSET 0.000000 0.000000 510.138000
        CHANNELS 3 Xrotation Yrotation Zrotation
        JOINT l_Acromioclavicular
        {
          OFFSET 0.000000 -116.680500 0.000000
          CHANNELS 3 Xrotation Yrotation Zrotation
          JOINT l_Shoulder
          {
            OFFSET 0.000000 -116.680500 -54.270000
            CHANNELS 3 Xrotation Yrotation Zrotation
            JOINT l_Elbow
            {
              OFFSET 0.000000 0.000000 222.507000
              CHANNELS 3 Xrotation Yrotation Zrotation
              JOINT l_Wrist
              {
                OFFSET 0.000000 0.000000 276.777000
                CHANNELS 3 Xrotation Yrotation Zrotation
                End Site
```

```
            {
              OFFSET 0.000000 0.000000 144.720000
            }
          }
        }
      }
    }
    JOINT r_Sternoclavicular
    {
      OFFSET 0.000000 0.000000 510.138000
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT r_Acromioclavicular
      {
        OFFSET 0.000000 116.680500 0.000000
        CHANNELS 3 Xrotation Yrotation Zrotation
        JOINT r_Shoulder
        {
          OFFSET 0.000000 116.680500 -54.270000
          CHANNELS 3 Xrotation Yrotation Zrotation
          JOINT r_Elbow
          {
            OFFSET 0.000000 0.000000 222.507000
            CHANNELS 3 Xrotation Yrotation Zrotation
            JOINT r_Wrist
            {
              OFFSET 0.000000 0.000000 276.777000
              CHANNELS 3 Xrotation Yrotation Zrotation
              End Site
              {
                OFFSET 0.000000 0.000000 144.720000
              }
            }
          }
        }
      }
    }
    JOINT Cervical
    {
      OFFSET 0.000000 0.000000 510.138000
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT Skullbase
      {
        OFFSET 0.000000 0.000000 72.360000
        CHANNELS 3 Xrotation Yrotation Zrotation
        End Site
        {
          OFFSET 0.000000 0.000000 180.900000
        }
      }
    }
  }
}
JOINT l_Hip
{
  OFFSET 0.000000 -104.922000 0.000000
  CHANNELS 3 Xrotation Yrotation Zrotation
  JOINT l_Knee
  {
    OFFSET 0.000000 0.000000 356.373000
    CHANNELS 3 Xrotation Yrotation Zrotation
    JOINT l_Ankle
    {
```

```
        OFFSET 0.000000 0.000000 356.373000
        CHANNELS 3 Xrotation Yrotation Zrotation
        End Site
        {
          OFFSET 0.000000 0.000000 217.080000
        }
      }
    }
  }
  JOINT r_Hip
  {
    OFFSET 0.000000 104.922000 0.000000
    CHANNELS 3 Xrotation Yrotation Zrotation
    JOINT r_Knee
    {
      OFFSET 0.000000 0.000000 356.373000
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT r_Ankle
      {
        OFFSET 0.000000 0.000000 356.373000
        CHANNELS 3 Xrotation Yrotation Zrotation
        End Site
        {
          OFFSET 0.000000 0.000000 217.080000
        }
      }
    }
  }
}
MOTION
Frames: 3
Frame Time: 0.033333
502.459493 257.333667 639.802870 1.700498 -6.882363 -179.594274 0.000000 0.000000
0.000000 -1.231736 -13.201805 -5.576928 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 93.372598 3.590559 66.544828 0.000000 50.342207 0.000000 0.000000 0.000000
86.790763 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 -80.820769 -7.061278
-55.134580 0.000000 51.418193 0.000000 0.000000 0.000000 63.027386 0.000000 39.021100
0.000000 0.000000 0.000000 0.000000 3.945488 -177.331499 0.000000 0.000000 -6.963592
0.000000 0.000000 -90.000000 0.000000 0.000000 180.000000 0.000000 0.000000 0.000000
0.000000 0.000000 -90.000000 0.000000

506.723461 258.048658 642.125385 1.485601 -7.320077 176.987280 0.000000 0.000000
0.000000 -0.393428 -13.465185 -1.290738 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 95.557280 0.625421 65.063712 0.000000 64.725860 0.000000 0.000000 0.000000
147.139337 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 -82.816133 -4.180210
-63.381642 0.000000 62.222464 0.000000 0.000000 0.000000 33.191067 0.000000 39.213716
0.000000 0.000000 0.000000 0.000000 3.827811 -178.890877 0.000000 0.000000 -6.229981
0.000000 0.000000 -90.000000 0.000000 0.000000 180.000000 0.000000 0.000000 0.000000
0.000000 0.000000 -90.000000 0.000000

500.506925 256.161101 639.997866 1.993995 -7.015444 -178.727495 0.000000 0.000000
0.000000 -0.341594 -13.827125 -5.092958 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 93.970188 -1.159427 63.905184 0.000000 76.854872 0.000000 0.000000 0.000000
147.647672 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 -84.141932 -2.510111
-66.834904 0.000000 74.249378 0.000000 0.000000 0.000000 36.529596 0.000000 39.716160
0.000000 0.000000 0.000000 0.000000 4.088937 -176.926080 0.000000 0.000000 -7.466625
0.000000 0.000000 -90.000000 0.000000 0.000000 180.000000 0.000000 0.000000 0.000000
0.000000 0.000000 -90.000000 0.000000

ATTRIBUTES
JOINT Root
{
  ELLIPTIC_CONE
```

```
  {
    GEOMETRY  154.175205  198.669223  177.645182  0.886304  1.000000
    OverLaping  0
  }
  DOF
  {
    Xposition  VAR  279483008  1.000000  1
    Yposition  VAR  279483336  1.000000  1
    Zposition  VAR  279483632  1.000000  1
    Xrotation  VAR  279505368  1.000000  1
    Yrotation  VAR  279562080  1.000000  1
    Zrotation  VAR  279529600  1.000000  1
    Cone_aBase  CST  279486208  1.000000  1
    Cone_bBase  CST  279486568  1.000000  1
    Cone_aTop  CST  279487008  1.000000  1
    Cone_height  CST  279494680  1.000000  1
  }
  TREE_XY
  {
    Position  249  24
  }
  FEATURE
  {
    Label  0
    Type  1
    LinkedBone  None
    LinkedFeature  -1
    X  CST  285147144  1.000000  -76.099622  1
    Y  CST  285147408  1.000000  140.492442  1
    Z  CST  285147672  1.000000  133.017606  1
    Target  1
    Weight  1
    Exported  1
  }
  FEATURE
  {
    Label  1
    Type  1
    LinkedBone  None
    LinkedFeature  -1
    X  CST  284597528  1.000000  -85.001207  1
    Y  CST  284597832  1.000000  -130.054355  1
    Z  CST  284598136  1.000000  142.580651  1
    Target  1
    Weight  1
    Exported  1
  }
  MKM
  {
    Name  Pelvis
    RoatationRef  0.000000  0.000000  0.000000
  }
}
```

Dans ce fichier, nous pouvons voir apparaître trois blocs : HIERARCHY, MOTION et ATTRIBUTES. Le premier bloc décrit le squelette avec l'ensemble des articulations et le placement relatif de ces dernières. Un élément du squelette est décrit par un champ :

```
JOINT Name
```

```
{
  OFFSET  A B C
  CHANNELS 3 Xrotation Yrotation Zrotation
  End Site {
  }
}
```

Le champ JOINT décrit le nom de l'articulation. On peut noter que la racine de la chaîne n'est pas un JOINT mais ROOT. Le champ OFFSET est la position du centre de rotation de l'articulation par rapport à l'articulation mère. CHANNELS est le nombre de degrés de liberté avec le nom de ces degrés de liberté. Ces noms sont standards (Xrotation, Yrotation, Zrotation, Xposition, Yposition, Zposition). Enfin, les articulations sommitales ont un champ additionnel (et optionnel) qui est le End Site. Ce dernier champ permet de fixer l'extrémité de la chaîne articulaire.

Le second bloc décrit le mouvement. Sur une ligne, nous trouvons l'ensemble des valeurs articulaires pour chaque articulation du squelette. Elles sont ordonnées de sorte à ce que la première valeur corresponde au premier champ de CHANNEL du Root, la seconde, le second champ, *etc.* la dernière valeur correspondant à la dernière valeur du dernier champ CHANNEL.

Enfin, le troisième bloc contient différents champs :

ELLIPTIC_CONE : permet de spécifier la primitive géométrique utilisée. Deux attributs existent : GEOMETRY (décrivant l'ensemble des dimensions du cône) et OverLaping permettant lors du dimensionnement d'autoriser la recherche de contours cachés (« derrière ») par cette partie du corps.

DOF : permet de spécifier les contraintes sur les articulations.

Xposition (similaire pour les cinq suivants) : Il s'agit de décrire le degrés de liberté. Celui-ci peut être laissé libre (VAR), être un paramètre constant (CST) c'est-à-dire estimé si nécessaire (cela s'applique par exemple pour la position des cuisses par rapport au bassin, la position des cuisses doit être estimée mais laissée constante lors de l'estimation du mouvement) ou encore fixe (FIX) au quel cas rien ne peut modifier la valeur associée. Le premier chiffre est un identifiant et permet de poser des contraintes de symétrie dans le squelette. Dans le squelette, si deux identifiants sont identiques alors les variables articulaires respectives réagiront de la même manière si l'une d'elle est modifiée. Le second chiffre est un coefficient d'échelle. Le dernier chiffre est un coefficient multiplicateur de la valeur angulaire (la symétrie peut nécessiter de faire varier les valeurs angulaires de manière opposée).

Cone_aBase (similaire pour les trois suivants) : La construction est similaire à la description ci-dessus.

TREE_XY : permet de faciliter le placement du diagramme représentant la hiérarchie du squelette dans le logiciel MVACTOR.

FEATURE : permet de décrire un point (comme un marqueur VICON) attaché à la primitive considérée. Il y a plusieurs champs :

Label : permet d'identifier le point

Type : un point quelconque (=0), ou un point reconstruit par l'utilisateur (=1) ou encore lié à un autre membre (=2) (pour créer des contraintes secondaires comme pour l'épaule).

LinkedBone : si type=2 alors c'est l'identifiant du membre auquel le point est aussi attaché

LinkedFeature : si type=2 alors c'est l'identifiant du feature auquel le point est aussi attaché

X : (similaire pour les deux suivants) même structure que pour Xposition avec un champ supplémentaire indiquant la valeur de la position (sur l'axe considéré).

Target : le feature peut être attaché au squelette ou à la surface *3D*.

Weight : poids du feature si celui-ci est utilisé lors d'une optimisation

Exported : Est-il exploitable pour une comparaison avec les données VICON ?

MKM : permet de décrire la correspondance entre les noms choisis par nos soins et ceux demandés par MKM. D'autre part, certaines articulations que nous utilisons sont incompatibles avec le logiciel MKM, nous pouvons donc choisir de ne pas les exporter dans le fichier destiné à l'échange.

C.2 Conversion vers la norme H-anim

Comme nous avons pu le voir dans le chapitre 4, nous avons choisi d'utiliser une convention d'orientation des repères dans la chaîne articulaire de sorte que les repères associés aux articulations aient l'axe $\boldsymbol{k}$ orienté selon l'axe principal du cône. Or, le logiciel utilisé par l'UHB (MKM) utilise la convention de type H-ANIM qui stipule que pour la position de repos l'ensemble des repères sont orientés de sorte qu'ils soient une translation du repère de référence. Nous avons donc dû établir la correspondance entre notre convention d'orientation et la norme H-ANIM. Nous allons décrire cette conversion dans la suite de ce paragraphe.

Supposons que la figure C.1-(a) représente la configuration au repos et donc la pose de référence de notre chaîne cinématique. Nous considérons que le repère de référence est celui associé à la racine de la chaîne. Dans la convention que nous utilisons actuellement, les valeurs des paramètres articulaires pour la pose de référence ne sont pas nulles. En effet, si l'ensemble des paramètres articulaires étaient à 0, alors nous aurions la configuration illustrée par la figure C.2 pour laquelle les repères associés à chacune des articulations sont alignés avec le repère de référence. L'objectif est de faire en sorte que pour la configuration de repos, l'ensemble des variables articulaires soit nul. Nous allons décrire les étapes nécessaires pour effectuer cette transformation.

Elle s'effectue en deux étapes :
- Modification de la description du squelette.
- Modification des valeurs articulaires pour le mouvement de la chaîne.

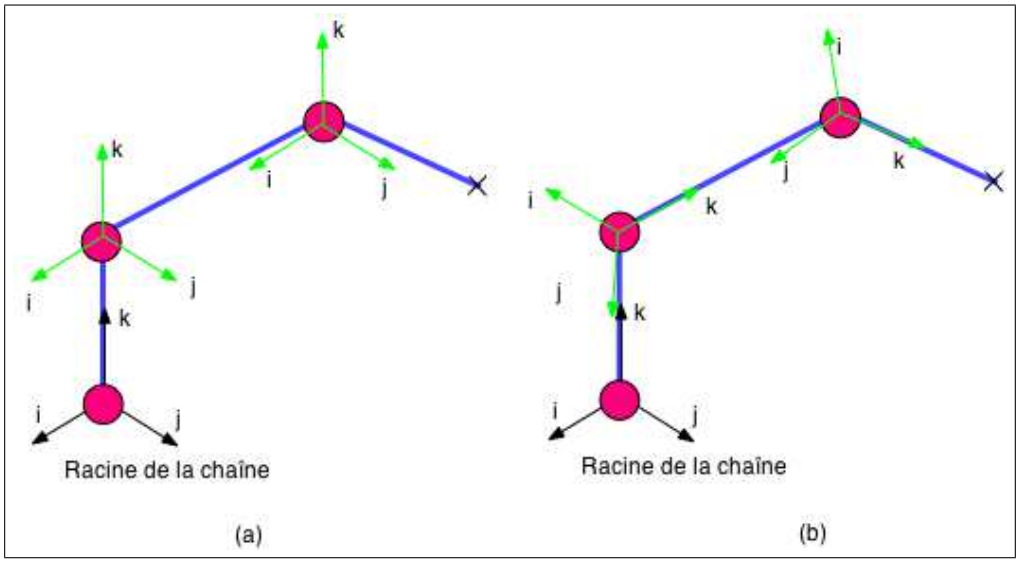


FIG. C.1: Convention d'orientation des repères pour la chaîne cinématique. (a) Dans notre convention, l'axe $\boldsymbol{k}$ est orienté vers l'articulation fille. (b) La convention H-Anim stipule qu'en position de référence, tous les repères sont alignés.

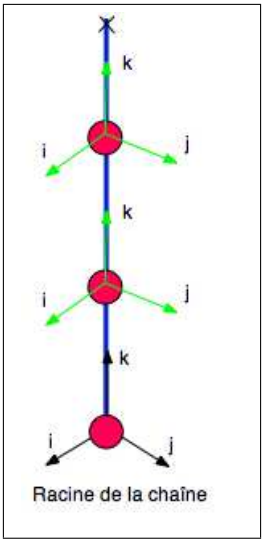


FIG. C.2: La position de référence (toutes les variables articulaires sont à 0, $\boldsymbol{\Lambda} = \mathbf{0}$) est illustrée ici.

C.2.1 Le squelette

Le premier bloc descriptif du fichier BVH doit être modifié. En effet, le placement relatif des articulations doit être fait de sorte que pour un vecteur de paramètres articulaires nuls, la pose de référence soit correcte.

Cette opération s'apparente à un changement de référentiel. En pratique, nous calculons les coordonnées de tous les centres articulaires dans le repère de référence (les coordonnées absolues). Quelle que soit la convention d'orientation utilisée, ces coordonnées absolues sont invariantes. Nous utilisons donc ces coordonnées pour effectuer le changement de référentiel.

Pour la pose de référence, dans la convention H-ANIM, les repères étant tous orientés de la même manière il suffit de calculer la position du centre articulaire d'une articulation dans le repère de référence, translaté sur le centre articulaire de l'articulation mère. Ce calcul s'écrit simplement :

$$\boldsymbol{X}_l = \boldsymbol{X}_{l-1}^{\mathcal{W}} - \boldsymbol{X}_l^{\mathcal{W}}, \tag{C.1}$$

où $\boldsymbol{X}_l$ est le vecteur des coordonnées relatives du centre articulaire de l'articulation l (par rapport à l'articulation mère) et $\boldsymbol{X}_{l-1}^{\mathcal{W}}$ le vecteur des coordonnées absolues de l'articulation mère.

C.2.2 Le mouvement

La modification du premier bloc implique la nécessité de modifier le second bloc décrivant le mouvement au cours de la séquence vidéo. L'objectif est de calculer l'orientation relative des segments pour la nouvelle convention en fonction des orientations dans notre convention. De la même manière que précédemment, l'orientation absolue des segments est la même quelle que soit la convention choisie. Nous nous servons donc de ces orientations absolues pour effectuer la conversion.

Plus précisément, l'objectif est de déterminer la rotation d'une articulation par rapport à une pose de référence. Pour cela, nous utilisons la modélisation en référence introduite dans le chapitre 3. Ainsi, toutes les rotations sont exprimées par rapport à la pose initiale.

Notons $\mathbf{M}_l$ la matrice d'orientation et de position d'un segment l par rapport à l'articulation mère pour la convention que nous utilisons. Si nous notons $\mathbf{M}_l^s$ la matrice d'orientation et de position pour la convention H-ANIM, nous avons :

$$\mathbf{M}_l^s = \mathbf{M}_0^0 \dots \mathbf{M}_{l-1}^0 \mathbf{M}_l (\mathbf{M}_0^0 \dots \mathbf{M}_l^0)^{-1}, \tag{C.2}$$

où $\mathbf{M}_l^0$ dénote la matrice de pose initiale de l'articulation l (pour notre convention).

Enfin, ce qui est stocké dans le fichier BVH, ce sont les valeurs des angles de Cardan. Il s'agit donc d'extraire les angles de la matrice $\mathbf{M}_l^s$ comme décrit dans l'annexe A.3.

Après conversion du fichier exemple donné précédemment, nous obtenons le fichier BVH suivant :

```
HIERARCHY
ROOT Root
{
  OFFSET 502.459493 257.333667 639.802870
  CHANNELS 6 Xposition Yposition Zposition Xrotation Yrotation Zrotation
  JOINT Sacroiliac
  {
    OFFSET -21.677471 -5.329513 179.517391
    CHANNELS 3 Xrotation Yrotation Zrotation
    JOINT Dorsal
    {
      OFFSET -4.335494 -1.065903 35.903478
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT l_Sternoclavicular
      {
        OFFSET 56.237605 -24.889805 506.417425
        CHANNELS 3 Xrotation Yrotation Zrotation
        JOINT l_Acromioclavicular
        {
          OFFSET 10.154272 116.147516 4.580879
          CHANNELS 3 Xrotation Yrotation Zrotation
          JOINT l_Shoulder
          {
            OFFSET 4.171548 118.795367 -49.293315
            CHANNELS 3 Xrotation Yrotation Zrotation
            JOINT l_Elbow
            {
              OFFSET 4.055674 222.453926 -2.677180
              CHANNELS 3 Xrotation Yrotation Zrotation
              JOINT l_Wrist
              {
                OFFSET -58.502591 180.176555 201.790396
                CHANNELS 3 Xrotation Yrotation Zrotation
                End Site
                {
                  OFFSET -30.589590 94.209963 105.511318
                }
              }
            }
          }
        }
      }
      JOINT r_Sternoclavicular
      {
        OFFSET 56.237605 -24.889805 506.417425
        CHANNELS 3 Xrotation Yrotation Zrotation
        JOINT r_Acromioclavicular
        {
          OFFSET -10.154272 -116.147516 -4.580879
          CHANNELS 3 Xrotation Yrotation Zrotation
          JOINT r_Shoulder
          {
            OFFSET -16.136996 -113.499664 -58.455074
            CHANNELS 3 Xrotation Yrotation Zrotation
            JOINT r_Elbow
            {
              OFFSET 11.994131 -220.959160 23.292817
              CHANNELS 3 Xrotation Yrotation Zrotation
              JOINT r_Wrist
              {
                OFFSET -91.478951 -156.753425 208.962851
                CHANNELS 3 Xrotation Yrotation Zrotation
                End Site
```

```
              {
                OFFSET -47.832131 -81.962575 109.261621
              }
            }
          }
        }
      }
      JOINT Cervical
      {
        OFFSET 56.237605 -24.889805 506.417425
        CHANNELS 3 Xrotation Yrotation Zrotation
        JOINT Skullbase
        {
          OFFSET -38.909323 0.995808 61.000349
          CHANNELS 3 Xrotation Yrotation Zrotation
          End Site
          {
            OFFSET -97.273306 2.489519 152.500873
          }
        }
      }
    }
  }
  JOINT l_Hip
  {
    OFFSET -0.737620 104.875805 3.024483
    CHANNELS 3 Xrotation Yrotation Zrotation
    JOINT l_Knee
    {
      OFFSET 59.201202 -13.962366 -351.143824
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT l_Ankle
      {
        OFFSET 16.158699 -14.090613 -355.727517
        CHANNELS 3 Xrotation Yrotation Zrotation
        End Site
        {
          OFFSET -216.856572 -0.122121 -9.845736
        }
      }
    }
  }
  JOINT r_Hip
  {
    OFFSET 0.737620 -104.875805 -3.024483
    CHANNELS 3 Xrotation Yrotation Zrotation
    JOINT r_Knee
    {
      OFFSET 42.704618 10.499142 -353.649260
      CHANNELS 3 Xrotation Yrotation Zrotation
      JOINT r_Ankle
      {
        OFFSET 42.704618 10.499142 -353.649260
        CHANNELS 3 Xrotation Yrotation Zrotation
        End Site
        {
          OFFSET -215.510378 -0.764595 -26.046473
        }
      }
    }
  }
}
```

```
MOTION
Frames: 3
Frame Time: 0.033333
502.459493 257.333667 639.802870 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -0.000000 0.000000 -0.000000

506.723461 258.048658 642.125385 0.197459 -0.349075 -3.402493 -0.000000 0.000000
-0.000000 -0.354803 0.068684 4.166190 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -2.601714 -1.325276 -2.675118 13.759717 0.299586 4.153857 -29.022886 29.200689
53.657735 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 1.669380 7.725432
-3.936495 -9.576314 -0.622858 -5.022076 13.196328 14.400501 -24.420164 -0.016775
-0.191735 -0.007590 -0.000000 0.000000 -0.000000 0.119202 1.558587 -0.047549 -0.000000
-0.733037 0.028991 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000
0.000000 -0.000000 -0.000000 0.000000 -0.000000

500.506925 256.161101 639.997866 0.189819 -0.159330 0.855941 -0.000000 0.000000
-0.000000 -0.835297 0.598742 0.404435 -0.000000 0.000000 -0.000000 -0.000000 0.000000
-0.000000 -1.401347 -2.602706 -4.623586 25.322864 1.314720 7.518692 -29.401989 29.308642
54.213615 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 3.193580 10.865136
-6.136674 -20.253958 -0.321854 -10.645774 11.391567 13.089490 -21.470439 -0.060656
-0.691870 -0.027655 -0.000000 0.000000 -0.000000 -0.142976 -0.405586 -0.000000 -0.000000
0.502639 -0.019885 -0.000000 0.000000 -0.000000 -0.000000 0.000000 -0.000000 -0.000000
0.000000 -0.000000 -0.000000 0.000000 -0.000000
```

Nous pouvons constater que pour la première image, les valeurs articulaires sont nulles. Nous avons décidé dans ce cas particulier de ne pas respecter le norme H-ANIM quant à la position de référence initiale du squelette humain. En effet, la norme H-ANIM stipule que le personnage doit être dressé, mains le long du corps avec les paumes tournées vers les hanches et le pieds serrés. Ici, le squelette à une pose quelconque pour la pose de référence.

Bibliographie

[1] 3DSMax. *http ://www.autodesk.com/3dsmax*.

[2] A. Agarwal and B. Triggs. Learning to track 3-D human motion from silhouettes. In *International Conference on Machine Learning*, pages 9–16, 2004.

[3] J. K. Aggarwal and Q. Cai. Human motion analysis : A review. *Computer Vision and Image Understanding*, 73(3) :428–440, 1999.

[4] J. Allard, J.-S. Franco, C. Ménier, E. Boyer, and B. Raffin. The grimage platform : A mixed reality environment for interactions. In *International Conference on Vision Systems*, page 46, 2006.

[5] H. Alt and M. Godau. Measuring the resemblance of polygonal curves. In *Symposium on Computational Geometry*, pages 102–109, 1992.

[6] E. Arnaud. *Méthodes de filtrage pour du suivi dans des séquences d'images - Application au suivi de points caractéristiques.* PhD thesis, Thèse de l'Université de Rennes 1, mention Traitement du Signal et Télécommunications, Novembre 2004.

[7] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2) :174–188, 2002.

[8] J. Arvo and D. Kirk. *An introduction to ray tracing*, chapter A survey of ray tracing acceleration techniques, pages 201–262. Academic Press Ltd., 1989.

[9] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *International Conference on Pattern Recognition*, volume 3, pages 627–632, 1996.

[10] A. O. Balan and M. J. Black. An adaptive appearance model approach for model-based articulated object tracking. In *Computer Vision and Pattern Recognition*, volume 1, pages 758–765, 2006.

[11] O. Bernier and P. Cheung-Mon-Chan. Real-time 3-D articulated pose tracking using particle filtering and belief propagation on factor graphs. In *British Machine Vision Conference*, volume 1, pages 27–46, 2006.

[12] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) :239–256, 1992.

[13] J. Bigun, G. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8) :775–790, 1991.

[14] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *IEEE International Conference on Computer Vision*, pages 231–236, 1993.

[15] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34(3) :344–371, 1986.

[16] E. Boyer. On using silhouettes for camera calibration. In *Asian Conference on Computer Vision*, volume 1, pages 1–10, 2006.

[17] M. Bray, P. Kohli, and P. Torr. Posecut : Simultaneous segmentation and 3d pose estimation of humans using dynamic graph cuts. In *European Conference on Computer Vision*, pages 642–655, 2006.

[18] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Computer Vision and Pattern Recognition*, page 568, Washington, DC, USA, 1997. IEEE Computer Society.

[19] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition*, pages 8–15, 1998.

[20] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3) :179–194, February - March 2004.

[21] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnorr. Real-time optic flow computation with variational methods. In *Computer Analysis of Images Patterns*, pages 222–229, 2003.

[22] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.

[23] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for non-linear problems. *IEE proceedings. Part F. Radar and signal processing*, 146 :2–7, 1999.

[24] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Trans. Graph.*, 22(3) :569–577, 2003.

[25] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Computer Vision and Pattern Recognition*, volume 1, pages 77–84, 2003.

[26] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time part i : Theory and algorithms. *International Journal of Computer Vision*, 62(3) :221 – 247, May 2005.

[27] K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette across time : Part ii : Applications to human modeling and markerless motion tracking. *International Journal of Computer Vision*, 63(3) :225 – 245, August 2005.

[28] K. M. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3d voxel reconstruction of human motions. In *Computer Vision and Pattern Recognition*, volume 2, pages 714 – 720, 2000.

[29] R. Cipolla and P. Giblin. *Visual motion of curves and surfaces*. Cambridge University Press, New York, NY, USA, 2000.

[30] I. Cohen, G. Medioni, and H. Gu. Inference of 3-D human body posture from multiple cameras for vision-based user interfaces. In *World Multi-Conference on Systemics, Cybernetics and Informatics*, 2001.

[31] F. Dagognet and J. Herman. *Etienne-Jules Marey : A Passion for the Trace*. Zone Books, 1992.

[32] P. Danielsson. Euclidean distance mapping. *Computer Vision, Graphics and Image Processing*, 14 :227–248, 1980.

[33] A. J. Davison, J. Deutscher, and I. Reid. Markerless motion capture of complex full-body movement for character animation. In *Eurographic Workshop on Computer Animation and Simulation*, pages 3–14. Springer-Verlag New York, Inc., 2001.

[34] Q. Delamarre and O. Faugeras. 3-D articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81 :328–357, 2001.

[35] D. Demirdjian. Enforcing constraints for human body tracking. In *Workshop on Multi-Object Tracking*, 2003.

[36] D. Demirdjian and T. Darell. 3-D articulated pose tracking for untethered diectic reference. In *Multimodal Interfaces*, pages 267–272, 2002.

[37] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *IEEE International Conference on Computer Vision*, volume 1, pages 1071–1078, 2003.

[38] J. Deutscher, A. Blake, and I. Reid. Articulated body motion capture by annealed particle filtering. In *Computer Vision and Pattern Recognition*, pages 2126–2133, 2000.

[39] G. Dewaele, F. Devernay, and R. Horaud. Hand motion from 3d point trajectories and a smooth surface model. In *European Conference on Computer Vision*, volume 1, pages 495–507, 2004.

[40] S. Di Zenzo. Note : A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33(1) :116–125, 1986.

[41] D. DiFranco, T.-J. Cham, and J. Rehg. Reconstruction of 3-D figure motion from 2d correspondences. In *Computer Vision and Pattern Recognition*, volume 1, pages 307–314, 2001.

[42] M. P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[43] P. Dollar, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Computer Vision and Pattern Recognition*, pages 1964–1971, 2006.

[44] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3) :197–208, 2000.

[45] T. Drummond and R. Cipolla. Real-time tracking of highly articulated structures in the presence of noisy measurements. In *IEEE International Conference on Computer Vision*, pages 315–320, 2001.

[46] P. Edward and J. Webb. Quaternions for computer vision and robotics. In *Computer Vision and Pattern Recognition*, pages 382–383, 1983.

[47] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann, December 2004.

[48] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Computer Vision and Pattern Recognition*, pages 66–73, 2000.

[49] S. Fisher and M. Lin. Fast penetration depth estimation for elastic bodies using deformed distance fields. *International Conference on Intelligent Robots and Systems*, 2001.

[50] D. Forsyth and J. Ponce. *Computer Vision – A Modern Approach*. Prentice Hall, New Jersey, 2003.

[51] A. Fuhrmann, G. Sobotka, and C. Gross. Distance fields for rapid collision detection in physically based modeling. In *GraphiCon*, pages 58–65, 2003.

[52] D. Gavrila and L. Davis. 3-D model-based tracking of humans in action : a multi-view approach. In *Computer Vision and Pattern Recognition*, pages 73–80, San Francisco CA, 1996.

[53] D. M. Gavrila. The visual analysis of human movement : A survey. *Computer Vision and Image Understanding*, 73(1) :82–98, 1999.

[54] J. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions Image Processing*, 12(8) :938–943, 2003.

[55] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, London, 1989.

[56] M. Gleicher and N. Ferrier. Evaluating video-based motion capture. In *Computer Animation*, pages 75–80, 2002.

[57] R. C. Gonzales and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2002.

[58] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE proceedings. Part F. Radar and signal processing*, 140(2) :107–113, 1993.

[59] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree : A hierarchical structure for rapid interference detection. *Computer Graphics*, 30(Annual Conference Series) :171–180, 1996.

[60] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. *cvpr*, 01 :220–227, 2004.

[61] A. Gray. *Modern Differential Geometry of Curves and Surfaces with Mathematica*, chapter Ruled Surfaces. Number 19. CRC Press, 2nd edition, 1993.

[62] R. Gross and J. Shi. The cmu motion of body (mobo) database. Technical Report CMU-RI-TR-01-18, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, June 2001.

[63] P. Grünwald. *Advances in Minimum Description Length : Theory and Applications*, chapter A Tutorial introduction to the minimum description length principle. MIT Press, 2005.

[64] P. Guigue and O. Devillers. Fast and robust triangle-triangle overlap test using orientation predicates. *Journal of Graphics Tools*, 8(1) :39–52, 2003.

[65] S. Guy and G. Debunne. Monte-carlo collision detection. Technical Report RR-5136, INRIA, March 2004.

[66] J. K. Hahn. Realistic animation of rigid bodies. In *SIGGRAPH*, pages 299–308, New York, NY, USA, 1988. ACM Press.

[67] Hanim. *http ://h-anim.org/*.

[68] S. Hasegawa and M. Sato. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. *Comput. Graph. Forum*, 23(3) :529–538, 2004.

[69] L. Herda, P. Fua, R. Plänkers, R. Boulic, and D. Thalmann. Using skeleton-based tracking to increase the reliability of optical motion capture. In *Human Movement Science*, volume 20, pages 313–341, 2001.

[70] L. Herda, R. Urtasun, and P. Fua. Hierarchical implicit surface joint limits for human body tracking. *Computer Vision and Image Understanding*, 99(2) :189–209, 2005.

[71] A. Hilton. Towards model-based capture of a persons shape, appearance and motion. In *Proceedings of the IEEE International Workshop on Modelling People*, September 1999.

[72] A. Hilton and P. Fua. Modeling people toward vision-based understanding of a person's shape, appearance and movement. *Comput. Vis. Image Underst.*, 81(3) :227–230, 2001.

[73] A. Hilton, P. Fua, and R. Ronfard. Modeling people : Vision-based understanding of a person's shape, appearance, movement, and behaviour. *Computer Vision and Image Understanding*, 103(2-3) :87–89, November 2006.

[74] K. E. Hoff, J. Keyser, M. Lin, D. Manocha, and T. Culver. Fast computation of generalized Voronoi diagrams using graphics hardware. *Computer Graphics*, 33 :277–286, 1999.

[75] D. Hogg. Model-based vision : A program to see a walking person. *Image and Vision Computing*, 1(1) :5–20, 1983.

[76] D. Huttenlocher, D. Klanderman, and J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9) :850–863, September 1993.

[77] A. Jaklic, A. Leonardis, and F. Solina. *Segmentation and Recovery of Superquadrics*, volume 20 of *Computational imaging and vision*. Kluwer, Dordrecth, 2000. ISBN 0-7923-6601-8.

[78] P. Jiménez, F. Thomas, and C. Torras. 3-D Collision Detection : A Survey. *Computers and Graphics*, 25(2) :269–285, 2001.

[79] S. Ju, M. J. Black, and Y. Yacoob. Cardboard people : A parameterized model of articulated motion. In *International Conference on Automatic Face and Gesture Recognition*, pages 38–44, 1996.

[80] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1453–1459, 2000.

[81] I. A. Kakadiaris and D. Metaxas. 3d human body model acquisition from multiple views. In *IEEE International Conference on Computer Vision*, pages 618–623, 1995.

[82] I. A. Kakadiaris and D. Metaxas. Three-dimensional human body model acquisition from multiple views. *International Journal of Computer Vision*, 30(3) :191–218, 1998.

[83] R. Kehl, M. Bray, and L. Van Gool. Full body tracking from multiple views using stochastic sampling. In *Computer Vision and Pattern Recognition*, pages 129–136, Washington, DC, USA, 2005. IEEE Computer Society.

[84] R. Kehl and L. Van Gool. Markerless tracking of complex human motions from multiple views. *Computer Vision and Image Understanding*, 103(2-3) :190–209, November 2006.

[85] D. Knossow, R. Ronfard, R. Horaud, and F. Devernay. Tracking with the kinematics of extremal contours. In *Asian Conference on Computer Vision*, pages 664–673, 2006.

[86] D. Knossow, J. van de Weijer, R. Horaud, and R. Ronfard. Articulated-body tracking through anisotropic edge detection. In *Workshop on Dynamical Vision, European Conference on Computer Vision*, May 2006.

[87] J. Koenderinck. *Solid Shape*. The MIT Press, 1990.

[88] J. Koenderink. What does the occluding contour tell us about solid shape ? *Perception*, 13 :321–330, 1984.

[89] J. J. Koenderink and A. J. van Doorn. Receptive field families. *Biological Cybernetics*, 63 :291–297, 1990.

[90] P. Kohli and P. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 922–929, 2005.

[91] R. Kulpa, F. Multon, and B. Arnaldi. Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum*, 24(3) :343–352, 2005.

[92] S. Lavalle. *Planning Algorithms*. Cambridge University Press, 2006.

[93] M. Lin and S. Gottschalk. Collision detection between geometric models : A survey. In *IMA Conference on Mathematics of Surfaces*, 1998.

[94] J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430) :567–576, 1995.

[95] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443) :1032–1044, 1998.

[96] G. Loy, M. Eriksson, J. Sullivan, and S. Carlsson. Monocular 3-D reconstruction of human motion in long action sequences. In *European Conference on Computer Vision*, volume 4, pages 442–455, 2004.

[97] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three dimensional shapes. In *Proceedings of the Royal Society of London*, volume 207, pages 187–216, 1978.

[98] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5) :530–549, 2004.

[99] F. Martin and R. Horaud. Multiple camera tracking of rigid objects. *International Journal of Robotics Research*, 21(2) :97–113, February 2002.

[100] Maya. *www.autodesk.com/alias*.

[101] A. Menache. *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers Inc., 1999.

[102] C. Ménier, E. Boyer, and B. Raffin. 3-D skeleton-based body pose recovery. In *Proceedings of the 3rd International Symposium on 3-D Data Processing, Visualization and Transmission*, 2006.

[103] I. Mikic, M. M. Trivedi, E. Hunter, and P. C. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3) :199–223, 2003.

[104] J. Mitchelson and A. Hilton. Simultaneous pose estimation of multiple people using multiple-view cues with hierarchical sampling. In *British Machine Vision Conference*, 2003.

[105] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2) :90–126, 2006.

[106] T. Möller. A fast triangle-triangle intersection test. *journal of graphics tools*, 2(2) :25–30, 1997.

[107] D. D. Morris and J. Rehg. Singularity analysis for articulated object tracking. In *Computer Vision and Pattern Recognition*, pages 289–296, 1998.

[108] MotionBuilder. *www.autodesk.com/motionbuilder*.

[109] R. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Ann Arbor, 1994.

[110] E. Muybridge. *Muybridge's Complete Human And Animal Locomotion : All 781 Plates from the 1887 Animal Locomotion*, volume 1. Dover Publications, 1979.

[111] E. Muybridge. *Muybridge's Complete Human And Animal Locomotion : All 781 Plates from the 1887 Animal Locomotion*, volume 2. Dover Publications, 1979.

[112] B. Naylor, J. Amanatides, and W. Thibault. Merging bsp trees yield polyhedral modeling results. In *SIGGRAPH*, pages 115–124, 1990.

[113] M. Niskanen, E. Boyer, and R. Horaud. Articulated motion capture from 3-D points and normals. In *British Machine Vision Conference*, volume 1, pages 439–448, 2005.

[114] J. F. O'Brien, B. Bodenheimer, G. Brostow, and J. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Graphics Interface*, pages 53–60, 2000.

[115] J. Ohya and F. Kishino. Human posture estimation from multiple images using genetic algorithm. In *International Conference on Pattern Recognition*, pages A :750–753, 1994.

[116] OpenGL. *http ://www.opengl.org*.

[117] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages 661–675, Copenhaguen, Denmark, June 2002.

[118] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of IEEE*, 92(3) :495–513, 2004.

[119] R. Plänkers and P. Fua. Articulated soft objects for multi-view shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10) :1182–1187, 2003.

[120] D. Ramanan and D. Forsyth. Finding and tracking people from the bottom up. In *Computer Vision and Pattern Recognition*, volume 2, pages 467–474, 2003.

[121] D. Ramanan, D. Forsyth, and A. Zisserman. Strike a pose : Tracking people by finding stylized poses. In *Computer Vision and Pattern Recognition*, volume 1, pages 271–278, 2005.

[122] S. Redon, Y. Kim, M. Lin, and D. Manocha. Fast continuous collision detection for articulated models. *In Proceedings of ACM Symposium on Solid Modeling and Applications*, 2004.

[123] J. Rehg, D. D. Morris, and T. Kanade. Ambiguities in visual tracking of articulated objects using two- and three-dimensional models. *International Journal of Robotics Research*, 22(6) :393 – 418, June 2003.

[124] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures : an application to human hand tracking. In *European Conference on Computer Vision*, pages 35–46, 1994.

[125] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *European Conference on Computer Vision*, volume 4, pages 700–714, June 2002.

[126] E. Rosten and T. Drummond. Rapid rendering of apparent contours of implicit surfaces for real-time tracking. In *British Machine Vision Conference*, volume 2, pages 719–728, 2003.

[127] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2) :461–464, 1978.

[128] A. Senior. Real-time articulated human body tracking using silhouette information. In *IEEE Workshop on Visual Surveillance/PETS*, October 2003.

[129] A. Shahrokni, V. Lepetit, and P. Fua. Bundle adjustment for markerless body tracking in monocular video sequences. In *ISPRS workshop on Visualization and Animation of Reality-based 3-D Models*, February 2003.

[130] K. Shoemake. Animating rotations with quaternion curves. *SIGGRAPH*, 19(3) :245–254, 1985.

[131] H. Sidenbladh and M. J. Black. Learning the statistics of people in images and video. *International Journal of Computer Vision*, 54(1-3) :183–209, 2003.

[132] L. Sigal, S. Bhatia, M. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *Computer Vision and Pattern Recognition*, volume 1, pages 421–428, 2004.

[133] L. Sigal and M. J. Black. Humaneva : Synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University, Department of Computer Science, September 2006.

[134] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black. Attractive people : Assembling loose-limbed models using non-parametric belief propagation. In *Advances in Neural Information Processing Systems*, pages 1539–1546, 2003.

[135] C. Sminchisescu. *Estimation algorithms for ambiguous visual models - Three Dimensional Human Modeling and Motion Reconstruction in Monocular Video Sequences*. PhD thesis, INPG, Juillet 2002.

[136] C. Sminchisescu and B. Triggs. Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 22(6) :371–379, 2003.

[137] B. Stenger. *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. PhD thesis, Department of Engineering, University of Cambridge, March 2004.

[138] A. Sundaresan and R. Chellappa. Markerless motion capture using multiple cameras. In *Computer Vision for Interactive and Intelligent Environment*, pages 15–26, 2005.

[139] R. Szeliski. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing : Image Understanding*, 58(1) :23–32, 1993.

[140] D. Thalmann, J. Shen, and E. Chauvineau. Fast realistic human body deformations for animation and VR applications. In *Computer Graphics International*, pages 166–174, 1996.

[141] C. Theobalt, M. Magnor, P. Schüler, and H.-P. Seidel. Combining 2d feature tracking and volume reconstruction for online video-based human motion capture. In *Pacific Conference on Computer Graphics and Applications*, page 96, 2002.

[142] P. Tresadern and I. Reid. Uncalibrated and unsynchronized human motion capture : A stereo factorization approach. In *Computer Vision and Pattern Recognition*, volume 1, pages 128–134, 2004.

[143] B. Triggs. *Reconstruction monoculaire du mouvement humain, et autres travaux 2000-2004*. Habilitation à diriger des recherches, Institut National Polytechnique de Grenoble, Grenoble, France, January 2005.

[144] B. Triggs and M. Sdika. Boundary conditions for young - van vliet recursive filtering. *IEEE Transactions on Signal Processing*, 54(6) :2365–2367, 2006.

[145] U. Usta. Comparison of quaternion and euler angle methods for joint angle animation of human figure models. Computer science master's thesis, Naval Postgraduate School, Monterey, California, March 1999.

[146] J. van de Weijer and T. Gevers. Tensor based feature detection for color images. In *IS&TSID's CIC*, 2004.

[147] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

[148] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multi-modality through mixture tracking. In *IEEE International Conference on Computer Vision*, Nice, France, June 2003.

[149] S. Wachter and H.-H. Nagel. Tracking persons in monocular image sequences. *Computer Vision and Image Understanding*, 74(3) :174–192, 1999.

[150] G. Welch and G. Bishop. An introduction to the kalman filter. Tech. Report 95-041, Univ. North Carolina, Chapel Hill, 2001.

[151] Z. Xu and M. Zhu. Color-based skin detection : survey and evaluation. In *Multi-Media Modelling Conference Proceedings*, pages 143–152, January 2006.

[152] I. T. Young and L. J. van Vliet. Recursive implementation of the gaussian filter, signal processing. *Signal Processing*, 44(2) :139–151, 1995.