HAL
open science

Paramétrisation de données de capture de mouvement pour la compression d'animations 3D

Maxime Tournier

To cite this version:

Maxime Tournier. Paramétrisation de données de capture de mouvement pour la compression d'animations 3D. Synthèse d'image et réalité virtuelle [cs.GR]. 2007. inria-00598400

HAL Id: inria-00598400

https://inria.hal.science/inria-00598400v1

Submitted on 6 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Paramétrisation de données de capture de mouvement pour la compression d'animations 3D

M2R Image Vision Robotique Juin 2007

Maxime Tournier

EVASION/LJK, INRIA Rhône-Alpes - ZIRST
655 Avenue de l'Europe
38334 Saint-Ismier CEDEX

Email : maxime.tournier@inrialpes.fr

Encadrant: Lionel Reveret

Email : lionel.reveret@inrialpes.fr

Résumé

Le présent document constitue le rapport d'un stage effectué de Mars à Juin 2007 dans le cadre du Master 2 Recherche intitulé « Image Vision Robotique »[1] de l'INPG[2].

Après une brève présentation du sujet, il décrit l'étude bibliographique, le travail effectué ainsi que les résultats obtenus, suivis d'une discussion de ces derniers ainsi que d'une ouverture sur de potentiels travaux futurs.

Le stage s'est déroulé dans l'équipe EVASION[3] du laboratoire LJK[4], dans les locaux de l'INRIA Rhône-Alpes.[5] L'encardrement était assuré par Lionel Reveret, à qui je transmets toute ma gratitude pour sa disponibilité et sa confiance.

Un grand merci également à toute l'équipe pour tous les bons moments passés en son sein.

Merci enfin aux auteurs du merveilleux logiciel TEXMACS[6], sans lequel je serais bien dans la mouise.

1. M2R-IVR, http://ivr.imag.fr/
2. Institut National Polytechnique de Grenoble, http://www.inpg.fr/
3. Équipe EVASION, http://evasion.inrialpes.fr/
4. Laboratoire Jean Kuntzmann, http://ljk.imag.fr/
5. Institut National de Recherche en Informatique et Automatique, site de Montbonnot St-Ismier, http://www.inrialpes.fr/
6. GNU TEXMACS, http://www.texmacs.org/

Table des matières

Résumé . . . 3

Table des matières . . . 5

1 Présentation . . . 7

1.1 Le mouvement humain . . . 7
1.2 Capture de mouvement . . . 7
1.3 Compression de données de mouvement . . . 9

2 État de l'art . . . 11

2.1 Compression d'animations . . . 11
2.2 Quantification de l'erreur . . . 12
2.3 Multirésolution . . . 13
2.4 Statistique non-linéaire . . . 14
2.5 Problèmes liés aux angles d'Euler . . . 14

3 Travail réalisé . . . 15

3.1 Objectifs . . . 15
3.2 Représentation des orientations . . . 15
3.3 Les quaternions - La sphère S^3 . . . 17
3.3.1 Historique - Définitions . . . 17
3.3.2 Exponentielle - Logarithme . . . 19
3.3.3 Espace tangent . . . 19
3.4 Représentation multi-résolution de données d'orientation . . . 21
3.4.1 Le *lifting scheme* . . . 21
3.4.2 Améliorations . . . 23
3.5 Analyse statistique de données d'orientation . . . 26
3.5.1 Moyenne de Karcher . . . 26
3.5.2 Analyse en géodésiques principales (AGP) . . . 28
3.5.3 Compression par AGP de données réelles . . . 29
3.6 Une méthode hybride . . . 32
3.7 Implémentation . . . 34

4 Bilan . . . 37

4.1 Résultats . . . 37
4.1.1 Animation de marche . . . 37
4.1.1.1 Compression faible . . . 38
4.1.1.2 Compression forte . . . 38
4.1.2 Animation de danse . . . 39
4.1.2.1 Compression faible . . . 39
4.1.2.2 Compression forte . . . 40
4.1.2.3 Compression très forte . . . 41
4.2 Discussion - Perspectives . . . 41
4.2.1 Compression . . . 42
4.2.2 Artéfacts visuels . . . 42
4.2.3 Préservation des discontinuités C^1 . . . 43
4.2.4 Analyse statistique . . . 43

Bibliographie . . . 45

Chapitre 1
Présentation

1.1 Le mouvement humain

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

Figure 1.1. Trois domaines nécéssitant des données de mouvement humain : médical, jeu vidéo, systèmes d'interaction.

1.2 Capture de mouvement

Ces données sont pour la plupart issues de la capture de mouvement (*motion capture*, ou *mocap*). Le principe en est le suivant : un acteur se vêtit d'une combinaison munie de capteurs de position et/ou d'orientations placées aux articulations, puis joue la scène à capturer. Les données des capteurs sont alors enregistrées, traitées, puis sauvegardées. Les trajectoires de ces capteurs peuvent alors être reconstruites ultérieurement, permettant ainsi de reconstituer le mouvement dans l'espace des articulations de l'acteur. Une version simplifiée du squelette de l'acteur est également stockée sous la forme d'un arbre dans le fichier de données, à raison d'un noeud par capteur. On dispose au final de données décrivant un squelette virtuel ainsi que le mouvement de chacune de ses articulations au cours du temps.

Figure 1.2. Un dispositif de capture de mouvement optique (*gauche*) et l'utilisation de données de mouvement pour l'animation d'un personnage virtuel (*droite*)

Plusieurs classes de systèmes de *motion capture* sont disponibles :

- Les dispositifs à marqueurs optiques passifs ou actifs, tels ceux de type Vicon. Les données sont souvent d'une grande qualité, mais des solutions aux problèmes d'occlusions doivent être envisagées. (*cf.* figure 1.2 gauche) Les données fournies dans ce cas sont uniquement des données de position.

- Les dispositifs éléctomagnétiques, qui permettent de s'affranchir des problèmes d'occlusion, mais dont les mesures sont souvent très bruitées. Ce type de système peut fournir des données de position/orientation, mais est cependant limité par la portée spatiale du champ, souvent faible.

- Les dispositifs mécaniques, tels les *datagloves* ou les exosquelettes, fournissent quant à eux des données de position/orientation très précises, mais peuvent s'avérer difficiles à mettre en oeuvre.

L'utilisation de ces systèmes coûte dans la plupart des cas très cher, aussi les bases de données de mouvements capturés constituent des ressources précieuses, d'autant que les données issues de *motion capture* sont à l'heure actuelles les meilleures en termes de qualité et de fidèlité au mouvement humain.

Les méthodes procédurales de synthèse de mouvement présentent souvent des résultats de qualité médiocre et/ou peu credibles, et les animations produites souffrent d'un aspect 'robotique' difficilement éliminable. Néanmoins, certaines méthodes statistiques récentes proposent de capturer les éléments clés d'un type d'animation, tels sa structure ou son style, par apprentissage sur de grandes bases de données. Le but de ces méthodes est de permettre autant que possible la réutilisation des données existantes afin de permettre l'édition, le mélange et l'adaptation de données de mouvement. Ces techniques donnent des résultats très encourageants, mais nécessitent elles-mêmes de grandes quantités de données de mouvement, issues pour la plupart de *motion capture*.

La capture de mouvement, on le voit, est à l'heure actuelle une technique incontournable pour l'obtention de données de mouvement de *qualité*. Elle est ainsi massivement utilisée dans tous les domaines ayant trait au mouvement humain, qu'il soit corporel ou facial (*cf.* figure 1.3).

Figure 1.3. L'utilisation de la capture du mouvement pour l'animation du personnage Gollum dans le film Le Seigneur des Anneaux (*gauche*), et pour le design de prothèses de la jambe (*droite*).

De telles quantités de données de mouvement presques brutes peuvent cependant poser problème en terme de taille de données, qu'il s'agisse du stockage ou de la transmission de celles-ci. On songe notemment aux jeux vidéo en ligne massivement multijoueurs, qui nécéssitent des quantités de bande passante gigantesques, ou plus simplement aux supports CD ou DVD traditionnels limités en taille de stockage.

Pour remédier à ce problème, il devient nécéssaire de recourir à des techniques de *compression* de données.

1.3 Compression de données de mouvement

Des techniques de compression standard sans pertes peuvent être utilisées, telles que *zip* ou *gzip*[1.1] mais présentent des facteurs de gain généralement faibles (de l'ordre de 5:1) et négligent totalement les connaissances *a priori* sur la nature des données à compresser, *i.e.* du mouvement humain. Il existe en effet des corrélations naturelles, telles les contraintes bio-mécaniques, dans le mouvement des articulations pour certains types de mouvements. Exploiter de telles redondances au sein d'*une* animation permet de compresser cette dernière. De même, les redondances existant au sein d'un *ensemble* d'animations peuvent être exploitées afin de compresser des bases de données entières d'animations.

Il peut également être intéressant de disposer de techniques de compression *avec perte*, qui présentent souvent des taux de compression très supérieurs, dans les cas où la conformité au mouvement originel n'est pas primordiale (*e.g.* pour une application aux niveaux de détails dans des jeux en ligne). Les données d'orientation possèdent cependant une structure complexe et la reconstruction à partir de données partielles peut s'avérer problématique. Aussi est-il essentiel de travailler avec une représentation solide des orientations afin d'assurer la qualité des animations reconstruites. Nous concentrerons notre travail sur ce dernier point.

Dans le cas d'une compression avec perte, il est en outre nécessaire de savoir évaluer ce qui peut être compressé tout en dégradant le moins possible la qualité de l'animation. Il faut par conséquent être capable de mesurer la qualité d'une animation, pour alors déterminer ce qui a, ou n'a pas, d'importance dans le mouvement. La notion même d'*importance* dans ce contexte est intimement liée à la perception que nous avons du mouvement, aussi la mise en place d'une mesure de qualité d'animation devra-elle vraisemblablement prendre en compte des études perceptives du mouvement humain.

La suite de ce document s'organise de la manière habituelle : après un état de l'art des techniques liées à la compression de données de mouvement, nous présentons le travail effectué, suivi d'une analyse critique des résultats obtenus ainsi que des perspectives en vue de futurs travaux.

1.1. gzip, http://www.gzip.org/

Chapitre 2

État de l'art

2.1 Compression d'animations

Le problème de la compression d'animations s'articule autour de deux axes principaux : la compression de *maillages* et celle de *squelettes*. Les travaux précédents traitent en grande majorité de la compression de maillages. En effet, la forte cohérence spatiale ainsi que temporelle que présente le mouvement des sommets d'un maillage au cours d'une animation rendent le domaine particulierement adapté à la compression. [Len99] détecte les portions de maillage dont le déplacement est rigide pour n'encoder que la transformation et les résidus. Une Analyse en Composantes Principales(ACP) peut également s'avérer efficace dans ce cas de données de dimension élevée. En utilisant cette approche, [SSK05] identifient les trajectoires de sommets couplées linéairement afin d'effectuer une compression en évincant les vecteurs propres non significatifs.

S'il semble plus difficile *a priori* de déceler de telles corrélations dans des données d'animations squelettiques, des travaux en biomécanique tels [Ale91] suggèrent cependant qu'un couplage existe bel et bien dans le mouvement des articulations pour certains types de mouvements humains. Il est alors possible de réduire la dimension de l'espace du mouvement par une technique appropriée.

[SHP04] effectuent une ACP sur un groupe de mouvements similaires afin de réduire la dimension de l'espace des configurations, et par la suite permettre la synthèse de mouvement par optimisation dans cet espace restreint. Les résultats de cette technique concernent cependant souvent des mouvements très courts, et l'apprentissage nécessite une base de nombreux mouvements similaires.

[GMHP04] proposent une technique de cinématique inverse en estimant la distribution de probabilité des poses pour un type de mouvement par apprentissage statistique. Cette distribution capture les différentes corrélations entre les mouvements des articulations du squelette du point de vue de la *structure* ainsi que du *style* du mouvement. La technique proposée repose cependant sur une méthode d'optimisation lourde.

[Ari06] utilise une ACP sur des mouvements du même type afin d'exhiber les corrélations dans les mouvement articulatoires, et permettre ainsi la compression *avec pertes* de bases de données de grande taille. Les taux de compression annoncés laissent à penser que la méthode utilisée détecte les corrélations avec succès. Cependant, cette technique est basée sur les positions des articulations, et les résultats dépendent donc fortement des topologies des squelettes utilisés lors de l'apprentissage. Il faut de plus appliquer un traitement post-opératoire d'optimisation au squelette pour s'assurer que les animations reconstruites possèdent des os de longueur fixe. Travailler sur les orientations des articulations permet de s'affranchir de telles contraintes.

La cohérence temporelle est également exploitée à des fins de compression. [Ari06] représente les trajectoires des articulations par des splines. Outre les problèmes liés à l'utilisation des positions mentionnés ci-dessus, l'échantillonnage des points de contrôle est également susceptible de simplifier le mouvement plus qu'il n'est souhaitable. [BPvdP07] adaptent la compression par ondelettes des données d'orientation en sélectionnant les éléments de la base afin de minimiser l'erreur quadratique de reconstruction des positions par rapport à l'original. La technique de compression employée dans ce cas n'est pas spécifiquement adaptée aux données d'orientations, et des problèmes liés à l'utilisation d'angles d'Euler sont à prévoir pour des compressions plus fortes.

Les taux de compression annoncés par [Ari06] et [BPvdP07] sont respectivement de 35:1 et 48:1. Il convient cependant de remarquer que ces taux sont obtenus en quantifiant les données sur 16 bits. Les taux de compression *intrinsèques* aux méthodes proposées sont donc deux fois moindres.

2.2 Quantification de l'erreur

Le problème de quantification de l'erreur commise en compressant reste relativement peu traité. La donnée d'une métrique d'erreur robuste sur les animations de personnages fait l'objet de quelques travaux récents. [RP03] propose une étude perceptive sur la question en se limitant aux mouvements ballistiques.

S'il est généralement admis que la norme quadratique constitue un indicateur assez faible de la qualité de l'animation reconstruite, peu de travaux proposent une métrique alternative analytique et efficace. [RPE+05] abordent la question de la quantification de l'aspect naturel d'une animation.

Ils distinguent 3 grandes classes de métriques :

- Un ensembles de règles heuristiques, qui pénalisent le score d'une animation si elles sont violées. La plupart de ces règles ont trait au respect des lois physiques et/ou des contraintes bio-mécaniques
- Des métriques *perceptives* mettant en évidence les principaux défauts remarqués par des utilisateurs
- Des métriques construites à partir de classifieurs entrainés sur une base de mouvements

Si les deux premières permettent souvent de détecter si un mouvement est physiquement correct, elles éprouvent beaucoup plus de difficultés quant à la quantification de l'aspect naturel, du style de l'animation.

La dernière, développée dans [RPE+05], s'appuie sur la supposition qu'un humain percevra un type de mouvement comme naturel lorsqu'il a déjà été vu de nombreuses fois, et qu'à l'inverse un mouvement inhabituel sera perçu comme non-naturel, la clé étant que les humains ont vu au cours de leur vie une *très grande* quantité de mouvements.

Cette classe de mesures détecte des styles naturels assez efficacement, mais souffre pourtant de plusieurs défauts: les mesures induites dépendent grandement de la base de mouvements utilisés pour entrainer les classificateurs, notemment en ce qui concerne les mouvements définis comme non-naturels. D'autre part, certaines anomalies physiques (en particulier locales) sont difficilement détectables car noyées dans le score global de l'animation.

Le problème de la mise en place d'une métrique d'erreur robuste reste donc un problème difficile et ouvert, que nous ne traiterons pas ici.

2.3 Multirésolution

Une des techniques les plus efficaces à l'heure actuelle dans le domaine de la compression d'images est l'analyse multi-résolution, notemment avec le format JPEG2000[2.1]. L'analyse multi-résolution d'une image, *e.g.* en base d'ondelettes, fournit une représentation en plusieurs niveaux de details, et permet de compresser l'image en supprimant les détails de faible importance. Un codage adapté permet alors de n'encoder que les détails restants. Cette technique offre d'excellents résultats du point de vue de la compression : les images sont de meilleure qualité à taille égale, en comparaison avec le format de compression d'images standard JPEG par exemple.

Figure 2.1. La traditionnelle Lena

Figure 2.2. La même, en JPEG2000 (100:1)

L'application de l'analyse multi-résolution aux données de capture de mouvement n'est cependant pas triviale : la structure et la topologie sphérique de l'espace des rotations SO(3) le placent hors du cadre de la théorie des ondelettes. S'il reste néanmoins possible d'appliquer une analyse en ondelette aux données angulaires (*i.e.* les angles d'Euler représentant la rotation), cette méthode hérite nécessairement des singularités de la représentation (telles le tristement célèbre *gimbal lock*, la périodicité et les discontinuités impliquées pour la corriger). De fait, des taux de compression élevés sont susceptibles de provoquer des résultats inattendus lors de la reconstruction.

Il est également possible de plonger la sphère des rotations dans un espace vectoriel de dimension supérieure (*e.g.* les matrices 3×3, ou l'espace des quaternions $\mathbb{H}$), d'effectuer une analayse en ondelettes dans cet espace puis de reprojeter le résultat sur la sphère. Mais là encore, des résultats inattendus peuvent se produire à mesure que les données reconstruites s'éloignent de la sphère, comme c'est le cas dans le problème de l'interpolation de rotations. [Gra98] propose une revue des différentes représentations des rotations avec leurs singularités, ainsi que les difficultés liées à l'utilisation de chacune.

Pour s'affranchir au maximum des problèmes inhérents à la représentation des orientations choisie, il semble raisonnable d'utiliser celles présentant le moins de singularités possibles, et d'effectuer l'analyse multi-résolution *dans cet espace*, dont la structure *n'est pas* linéaire. Si la théorie classique des ondelettes (utilisant la transformée de Fourier, dites de 1ère génération [Swe98]) se prête mal au cadre non-linéaire, il est néanmoins possible d'adapter le *lifting scheme* [Swe98] des ondelettes dites de 2ème génération aux données d'orientations.

Sur ce principe, [LS01] proposent une analyse multi-résolution de donnés d'orientation afin de permettre l'édition, le mélange ainsi que le recollement d'animations. L'application à la compression est mentionnée, bien que non traitée. Le schéma de subdivision employé est du type *Lazy Wavelets* [Swe98], avec un opérateur de prédiction utilisant les interpolations linéaires sphériques, ou SLERP [Sho85]. Les détails restants utilisés pour corriger les prédictions sont stockés sous forme de logarithmes de quaternions.

2.1. Format JPEG2000, http://www.jpeg.org/jpeg2000/

[RDS+05] proposent une généralisation de ce schéma aux variétés riemanniennes[2.2] en utilisant l'espace tangent. Les applications pratiques sont toutefois restreintes à certains groupes de Lie[2.3], pour lesquels le passage à l'espace tangent est facilement calculable *via* l'application exponentielle. Une application à la compression de données d'orientation d'aviation est présentée, avec des résultats très prometteurs.

2.4 Statistique non-linéaire

Comme mentionné plus haut, l'analyse statistique de données de dimension élevée permet souvent d'exhiber les redondances statistiques présentes dans les données, puis de les exploiter afin de permettre la compression. Mais là encore, les techniques classiques d'analyse statistique traitent en tres large majorité du cadre linéaire, et leur application aux données d'orientation souffre encore une fois des problèmes liés à la structure non-linéaire de l'espace des rotations, ainsi que des singularités des représentations.

Les travaux traitant de statistique non-linéaire sont relativement récents. Bon nombre de résultats triviaux dans le cadre linéaire ne sont alors plus valides. Le simple problème du calcul de la valeur moyenne de données de la sphère [BF01] ne peut alors plus s'exprimer en termes d'espérance probabiliste, mais se formule comme un problème de minimisation de distances géodésiques. Le calcul de la valeur moyenne de rotations, *i.e.* de données de la sphère S^3 entre dans ce cadre [Moa02].

En restreignant le cadre d'étude aux variétés riemanniennes [Pen06] donne un ensemble d'outils de base pour les probabilités et les statistiques, et propose des définitions telles que, entre autres, valeur moyenne, matrice de covariance et loi normale pour des données dans de tels espaces. Pennec rappelle également les conditions d'existence et d'unicité ([Kar77],[Ken90]) de la valeur moyenne concernant la répartition des données.

Fletcher propose un analogue de l'Analyse en Composantes Principales (ACP) dans le cadre des groupes de Lie [FLJ03], puis des variétés riemanniennes [FLPJ04]. La technique, nommée Analyse en *Géodésiques* Principales (AGP), consiste comme dans le cas Euclidien à projeter les données sur des axes en maximisant la variance. Fletcher propose en outre une approximation de l'AGP par une ACP dans l'espace tangent à la valeur moyenne des données. Il propose une application à l'étude de la variabilité des formes d'hippocampes[2.4] en relation avec la schizophrénie [FLPJ04].

2.5 Problèmes liés aux angles d'Euler

Du point de vue de la compression, les représentations les plus compactes de données d'orientation sont les angles d'Euler et les logarithmes de quaternions. Bien que les problèmes liés à l'utilisation des premiers sont connus et critiqués depuis longtemps [Gra98], ils demeurent les seuls (à notre connaissance) outils envisagés dans le cadre de la compression de données d'orientation.

À la lumière des travaux mentionnés ci-dessus dans le domaine de l'analyse multi-résolution et de la statistique non-linéaire, nous proposons une méthode de compression d'animations utilisant les logarithmes de quaternions exempte des défauts dont font preuve les angles d'Euler.

2.2. Une variété différentiable M est dite Riemanniene lorsque son espace tangent possède un produit scalaire, ce dernier variant continûement le long de la variété. Ce produit scalaire induit un élément de norme infinitésimale permettant la définition d'une métrique sur la variété.

2.3. Un groupe de Lie est une variété différentielle muni d'une structure de groupe, dont les operations de multiplication et d'inversion sont différentiables. Il formalise la notion intuitive de "groupe continu".

2.4. L'hippocampe dont il est question ici, ou corne d'Ammon, est une partie du cerveau humain dont le rôle serait lié à la mémoire d'événements à long terme.

Chapitre 3
Travail réalisé

3.1 Objectifs

Dans le but d'effectuer la compression de données d'orientation issues de capture du mouvement, nous envisageons l'utilisation des logarithmes de quaternions au lieu des traditionnels angles d'Euler.

Les méthodes classiques appliquées à ces derniers négligent totalement la structure non-linéaire de l'espace des rotations, et peuvent à tout moment produire des résultats inattendus, *a fortiori* dans le cas d'une forte compression des données avec perte.

Afin de prévenir ces défauts inhérents à la représentation des angles d'Euler, nous proposons une méthode de compression de ces données utilisant les logarithmes de quaternions s'appuyant sur des méthodes multi-résolutions ainsi que statistiques adaptées à la structure et à la topologie de l'espace des données d'orientation.

3.2 Représentation des orientations

L'orientation d'un solide dans l'espace à trois dimensions peut se définir par sa rotation par rapport à une configuration choisie comme réference. On représentera ainsi l'orientation d'un solide (ici un élement de squelette) par cette rotation.

Les rotations de l'espace ont été largement étudiées par le passé. Leur ensemble muni de la composition des rotations forme un groupe, sous-groupe des automorphismes orthogonaux de $\mathbb{R}^3$.

Une rotation peut se définir par son axe (une droite vectorielle orientée) ainsi qu'un paramètre angulaire. On remarque facilement que selon l'orientation de l'axe choisie, on obtient la même rotation en choisissant l'angle opposé. Cette ambigûité peut s'avérer gênante lors du traitement des données selon la paramétrisation choisie.

L'espace des rotations constitue une variété de dimension 3, et possède en outre une topologie sphérique (*cf.* partie 6). Ceci implique que toute paramétrisation d'une rotation par un vecteur de $\mathbb{R}^3$ présentera *nécéssairement* des singularités [Gra98], les deux espaces n'étant pas homéomorphes.

On trouve une revue des principales représentations dans [Gra98], que nous reprenons brièvement ci-dessous, avec leur principaux atouts et limitations. Des données de SO(3), le groupe des rotations de $\mathbb{R}^3$, peuvent se représenter par une matrice de rotation, des triplet d'angle d'Euler, des quaternions unitaires (*i.e.* la sphère S^3), eux-mêmes étant représentables par leurs logarithmes. Il apparaît que les représentations les plus compactes des données d'orientation sont les angles d'Euler, les logarithmes de quaternions et, dans une moindre mesure, les quaternions eux-mêmes.

Les angles d'Euler sont largement utilisés dans le domaine de l'animation, principalement en raison de leur aspect intuitif : étant donné un triplet d'angles selon 3 axes, il est relativement aisé de prevoir l'allure de la rotation résultante. Cependant, ils souffrent de problèmes majeurs bien connus : le *gimbal lock*, la périodicité, et les problèmes d'interpolation.

Le premier se traduit par une perte d'un degré de liberté lorsque le deuxième angle provoque l'alignement de deux des axes de rotation. Il en résulte des artefacts visuels lors de la lecture d'une animation. Ce défaut peut être corrigé en évitant le voisinage des angles posant problème, et en choisissant un autre triplet d'angles représentant la même orientation.

Le deuxième oblige à reistreindre les valeurs des angles dans l'intervalle $[-\pi, \pi]$, ce qui occasionne des discontinuités et pénalise l'analyse statistique des données.

Le troisième défaut est plus gênant dans la mesure où le plus court chemin entre deux triplets d'angles ne correspond pas au plus court chemin dans l'espace des rotations. Dans des cas critiques, deux triplets d'angles voisins se traduiront par des orientations très différentes. Pour une application à la compression, par laquelle le signal original est reconstruit approximativement, ce défaut est extrêmement gênant car les données reconstruites peuvent alors différer totalement de celles de départ et ce même pour un faible taux de compression.

L'utilisation des quaternions de la sphère unité permet de s'affranchir de ces défauts, même si l'abigûité de représentation d'une orientation entre un quaternion et son opposé subsiste. Le problème majeur dans le cas de la compression consiste à maintenir l'appartenance des données reconstruites à la sphère unité. La technique habituelle consiste à considérer l'espace vectoriel des quaternions tout entier, puis à reprojeter le résultat sur la sphère. Cette technique ne donne malheureusement de bons résultats que pour des quaternions proches.

Enfin, les logarithmes de quaternions, plus compacts, permettent de s'affranchir de la contrainte d'appartenance à la sphère unité. Ils sont relativement intuitifs à manipuler et sont facilement convertibles en quaternions et *vice versa*. La représentation est périodique, mais toutes les périodes sont portées par la même droite vectorielle. Leur inconvenient majeur est qu'il n'existe pas de formule analytique exacte simple traduisant la composition des rotations pour les logarithmes de quaternions, et leur interpolation pose problème pour des axes de directions très différentes.

Nous donnons à présent quelques brefs rappels sur les quaternions et leurs logarithmes, et présentons une méthode d'analyse multi-résolution pour les données d'orientation.

3.3 Les quaternions - La sphère S^3

3.3.1 Historique - Définitions

Les quaternions ont été découverts par Sir William Rowan Hamilton en 1843, qui cherchait une extension des nombres complexes permettant d'effectuer des multiplications et des divisions de vecteurs de $\mathbb{R}^3$. On trouve une définition et les propriétés des quaternions dans la littérature classique, cependant nous rappelons brievement leur construction ainsi que quelques propriétés.

De la même manière que l'on obtient le corps des complexes $\mathbb{C}$ en considérant les matrices réelles 2×2 de la forme $\begin{pmatrix} x & y \\ -y & x \end{pmatrix}$, on obtient le corps (non-commutatif) des quaternions $\mathbb{H}$ en considérant les matrices complexes 2×2 de la même forme. C'est également un $\mathbb{R}$-espace vectoriel de dimension 4, dont les vecteurs de base $1, i, j, k$ vérifient les relations :

$$i^2 = j^2 = k^2 = i.j.k = -1$$

Le conjugué d'un quaternion $q = (a, b, c, d)$ est défini comme : $\bar{q} = (a, -b, -c, -d)$. Tout comme pour les complexes, on a:

$$|q|^2 = q.\bar{q} = a^2 + b^2 + c^2 + d^2$$

Il s'ensuit que pour tout quaternion $q \neq 0$, on a : $q^{-1} = \frac{\bar{q}}{|q|}$. En particulier, pour les quaternions de la sphère unité, $q^{-1} = \bar{q}$.

On définit la partie imaginaire d'un quaternion comme $\frac{q - \bar{q}}{2}$, et sa partie réelle par $\frac{q + \bar{q}}{2}$. Les quaternions dont la partie réelle (resp. imaginaire) est nulle sont appellés quaternions réels (resp. imaginaires) purs.

On peut alors définir dans $\mathbb{H}$ l'opération de conjugaison par un quaternion q : $x \longmapsto q.x.\bar{q}$. Si q est unitaire, on remarque que :

1. L'espace des quaternions imaginaires purs, que l'on peut identifier à $\mathbb{R}^3$, est stable par cette opération.

2. L'application de conjugaison par un quaternion unitaire est linéaire en dimension finie, et conserve la norme : son déterminant est donc +1 ou -1.

En considérant l'application qui à un quaternion unitaire associe le déterminant (une fonction polynômiale, donc continue) de la conjugaison par ce quaternion, et par connexité de la sphère unité, la seule valeur[3.1] que peut prendre ce déterminant sur l'*ensemble* de la sphère est +1 (car la conjugaison par $1 \in S^3$ n'est autre que l'identité, de déterminant +1).

On conclut ainsi à l'équivalence entre conjugaison par un quaternion unitaire et rotation pour les vecteurs de $\mathbb{R}^3$. La correspondance entre axe/angle et quaternion unitaire est donnée par :

$$\forall \theta \in \mathbb{R}, \forall v \in \mathbb{R}^3, \qquad q_{\theta,v} = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right).\frac{v}{|v|}\right) \tag{3.1}$$

On peut ainsi représenter n'importe quelle rotation de $\mathbb{R}^3$ par un quaternion unitaire. L'ensemble des quaternions unitaires forme ainsi une sphère de dimension 3. La projection en dimension 3 de cette sphère est représentée figure 3.1.

3.1. Une application continue sur un connexe par arcs dans un ensemble discret est constante. S^3 étant connexe par arcs, l'application définie est donc constante sur toute la sphère.

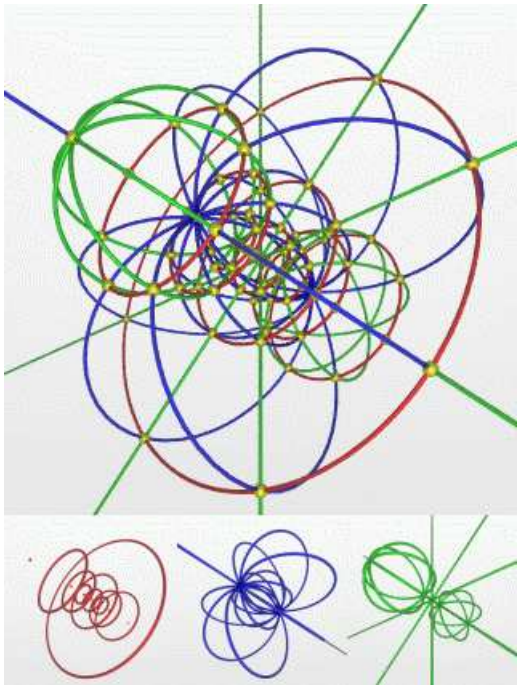


Figure 3.1. La projection stereographique des parallèles (rouge), méridiens (bleu), et hyperméridiens de la 3-sphère. *source: Wikipedia*

De la même manière qu'il existe dans la sphère usuelle S^2 deux directions différentes pour se déplacer (selon les *méridiens* et selon les *parallèles)*, il existe dans S^3 trois directions différentes pour se déplacer. La troisième direction constitue les "*hyperméridiens*".

La sphère S^3 constitue ce que l'on appelle une variété différentielle de dimension 3, ce qui signifie que l'on est en mesure de fournir un atlas de cartes de dimension 3 la décrivant, comme nous le faisons pour la sphère usuelle avec des cartes planes.

Intuitivement, une variété topologique de dimension n est un espace abstrait qui "ressemble" *localement* à l'espace euclidien usuel $\mathbb{R}^n$. Cependant, aucune information n'est donnée *a priori* quant à une eventuelle structure *globale*. La ressemblance locale avec $\mathbb{R}^n$ est la donnée de cartes permettant de décrire *localement* la variété. Pour qu'une variété soit qualifiée de *différentiable*, il faut en outre que l'opération consistant à passer d'une carte à une autre possède des propriétés de régularité suffisantes. La définition formelle est donnée ci-dessous :

Définition 3.1. *Variété différentielle*

Une variété différentielle de dimension $n \geqslant 1$ est un espace topologique séparé[3.2] tel que chacun de ses points $x \in M$ admet un voisinage ouvert U_x homéomorphe[3.3] à un ouvert de $\mathbb{R}^n$.

Si l'on nomme $\varphi_x : U_x \to V_x \in \mathbb{R}^n$ l'homéomorphisme associé à x, (U_x, φ_x) constitue une carte locale de M. Des cartes recouvrant entièrement M constituent un atlas de M.

La variété est dite différentielle de classe $\mathcal{C}^k$, avec $k \geqslant 1$, si pour tout couple (x, y) de voisinages (U_x, U_y) disjoints, la composée[3.4] $\varphi_x \circ \varphi_y^{-1}$ est un difféomorphisme[3.5] de classe $\mathcal{C}^k$.

La multiplication des quaternions est une opération différentiable, au même titre que l'inversion. Ces opérations munissent en outre S^3 d'une structure de groupe. La variété différentielle S^3 munie de ces opérations de groupe *différentiables* constitue par définition un groupe de Lie, qui formalise intuitivement la notion de "groupe continu".

3.2. Espace topologique séparé: espace topologique pour lequel deux points distincts admettent des voisinages disjoints

3.3. Un homéomorphisme est une bijection continue d'inverse continue.

3.4. La composée en question n'est autre que l'application "changement de carte", de $\mathbb{R}^n$ dans $\mathbb{R}^n$.

3.5. Un difféomorphisme de classe $\mathcal{C}^k$ est une application de classe $\mathcal{C}^k$ bijective, de réciproque $\mathcal{C}^k$ également.

3.3.2 Exponentielle - Logarithme

De la même manière qu'il est possible de représenter un nombre complexe $c \in \mathbb{C}$ par sa notation polaire $c = r.e^{i\theta}$ où $(r, \theta) \in \mathbb{R}^2$, il est également possible de représenter un quaternion $q \in \mathbb{H}$ de la manière suivante, où $(\theta, r) \in \mathbb{R}^2$ et v est un quaternion tel que $v^2 = -1$:

$$q = r.e^{v\frac{\theta}{2}} = r.\sum_{n \in \mathbb{N}} \frac{\left(\frac{\theta}{2}v\right)^n}{n!}$$

La condition $v^2 = -1$ est équivalente à ce que v soit un quaternion imaginaire pur normé, que l'on peut identifier à un vecteur normé de $\mathbb{R}^3$. Notons cependant que comme dans le cas des complexes, la représentation est 2π-périodique :

$$\forall k \in \mathbb{Z}, q = e^{v\left(\frac{\theta}{2} + 2k\pi\right)}$$

Ceci implique en particulier que $\forall v \in \mathbb{R}^3, |v| = 2\pi \Rightarrow e^v = 1$, et $|v| = \pi \Rightarrow e^v = -1$. En restreignant les valeurs de l'angle $\frac{\theta}{2}$ à l'intervalle $]-\pi, \pi[$, on peut néanmoins associer à tout quaternion $q \neq -1$ un unique vecteur de $\mathbb{R}^3$, et *vice-versa*. Le vecteur v ainsi que l'angle θ sont précisément ceux donnés dans l'équation 3.1, si bien que l'on a :

$$q = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right).v\right) = e^{v\frac{\theta}{2}}$$

Nous pouvons alors définir le logarithme d'un quaternion de la manière suivante :

$$\forall q \in S^3 - \{-1\}, \ln(q) = \ln\left(e^{v\frac{\theta}{2}}\right) = \frac{\theta}{2}v$$

Le fait que -1 ne soit pas représentable de manière univoque n'est pas gênant dans la mesure ou il correspond à la même rotation que le quaternion 1, soit l'identité.

3.3.3 Espace tangent

L'espace tangent à un point p d'une variété différentiable M est dans le cas général un espace vectoriel constitué des vecteurs tangents à la variété. Sa définition formelle est lourde[3.6] et sera omise ici, on peut cependant la trouver dans la littérature classique [DK00]. Il est noté $T_p(M)$ et formalise la notion intuitive de "direction" passant par p dans la variété :

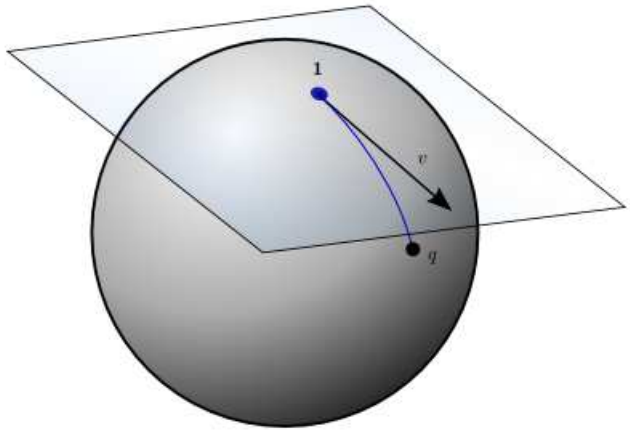


Figure 3.2. L'espace tangent à la sphère S^2, $T_1(S^2)$

3.6. Notons simplement que la difficulté vient du fait qu'on ne considère pas la variété comme "baignant" dans un espace de dimension supérieure, par exemple $\mathbb{R}^n$.

La définition d'un vecteur tangent à une variété différentiable en un point se fait uniquement en termes d'éléments de la variété, par exemple comme classe d'équivalence de courbes tangentes (dans une carte locale) en un point.

Le fait que S^3 possède une structure de groupe de Lie permet de lui associer naturellement une algèbre de Lie, qui à son tour s'identifie à l'espace tangent de la variété à l'identité (l'élément 1) [DK00]. La fonction exponentielle fournit en fait un passage de l'algèbre de Lie au groupe de Lie donc, de manière équivalente, de l'espace tangent au point identité vers la variété.

Comme nous l'avons vu plus haut lors de l'étude de l'application exponentielle, l'algèbre de Lie associée à S^3 s'identifie ainsi à l'ensemble des quaternions imaginaires purs, que l'on peut lui-même identifier à $\mathbb{R}^3$. L'espace tangent à l'identité de S^3, est donc $\mathbb{R}^3$.

Par symétrie de la sphère S^3, il est possible de passer à l'espace tangent à un point p différent de l'identité en faisant "rouler" la sphère sur son espace tangent de manière à aligner p avec l'identité :

$$\log_p(q) = \ln(p^{-1}.q) \tag{3.2}$$

Ceci permet en outre de définir une distance sur la sphère S^3, induite par la norme Euclidienne dans les espaces tangents :

$$\forall q_1, q_2 \in S^3, \mathrm{dist}(q_1, q_2) = \|\log_{q_1}(q_2)\|_2 = \|\log_{q_2}(q_1)\|_2 \tag{3.3}$$

Une autre propriété importante est que les sous-groupes à un paramètre d'un groupe de Lie correspondent aux droites vectorielles de son algèbre de Lie. Or ces sous-groupes constituent dans le cas de S^3 les géodésiques pour la norme définie ci-dessus, *i.e.* les plus courts chemins entre les points de S^3 au sens de cette norme.

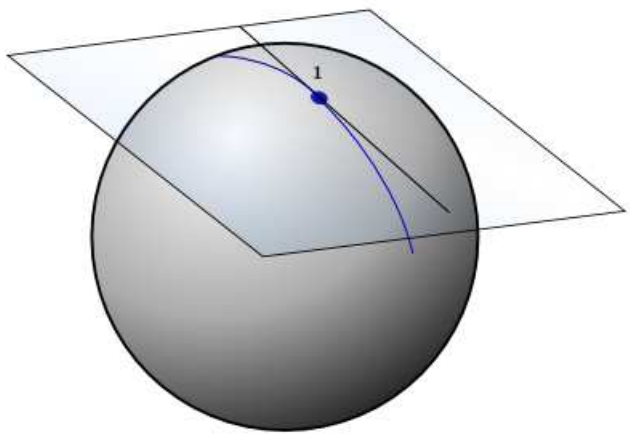


Figure 3.3. La courbe géodésique correspondant à une droite vectorielle de l'espace tangent

Concrêtement, tout cela signifie que l'on dispose d'une formule analytique pour parcourir la géodésique c_{q_1,q_2} reliant q_1 à q_2 :

$$\forall t \in [0,1], c_{q_1,q_2}(t) = q_1.e^{t.\ln\left(q_1^{-1}.q_2\right)} = q_1.(q_1^{-1}.q_2)^t$$

C'est de cette manière que s'effectue l'interpolation sphérique des quaternions ou SLERP: il s'agit d'une interpolation linéaire des quaternions dans l'espace tangent, garantissant le plus court chemin (*i.e.* le suivi d'une géodésique) entre les deux extrémites. Interpoler de cette manière possède des propriétés intéressantes dans le cadre de l'animation, comme notemment le fait que l'interpolation selon une géodésique s'effectue (par définition des géodésiques) à vitesse angulaire constante, et selon un plus court chemin.

Après ce rapide tour d'horizon de la sphère des quaternions unitaires, nous pouvons désormais présenter une méthode d'analyse multi-résolution ainsi que des outils statistiques adaptés à la structure particulière de cet espace.

3.4 Représentation multi-résolution de données d'orientation

3.4.1 Le *lifting scheme*

La méthode d'analyse multi-résolution d'animations proposée par Lee & Shin dans [LS01] est en réalité une version simplifiée du *lifting scheme* [Swe98], dont le principe est le suivant.

Soit S_0 un k-uplet de quaternions unitaires représantant le signal original au cours du temps. Pour simplifier, on considérera que le nombre d'échantillons k est une puissance de 2.

Le signal original S_0 est alors sous-échantillonné en ne gardant qu'un échantillon sur deux et constitue le signal S_1. Les données de S_0 ne se trouvant pas dans S_1 constituent l'ensemble U_0.

Le principe est alors d'utiliser les données de S_1 pour *prédire* celles de S_0, et donc celles de U_0. On dispose à cet effet d'un opérateur de prédiction, **pred**, dans le but de reconstruire au mieux le signal S_0 à partir de S_1.

Afin de corriger les erreurs de prédiction, on utilise les données de U_0. Les corrections à apporter au signal prédit $\tilde{S}_0$ en fonction des données de U_0 constitue l'ensemble des *détails* D_0. Une description schématique des différents "acteurs" de la méthode est donnée en figure 3.4.

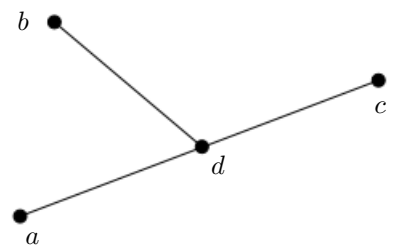


Figure 3.4. Un exemple du *lifting scheme*:
Soit $S_0 = \{a, b, c\}$. En sous-échantillonnant, nous formons $S_1 = \{a, c\}$, puis $U_0 = S_0 \backslash S_1 = \{b\}$.
La prédiction $\tilde{S}_0$ de S_0 en fonction de S_1 est $\{a, d, c\}$ où $d = \frac{a+c}{2}$.
L'ensemble des détails D_0 permettant de corriger la prédiction est alors le singleton $\{b - d\}$.

Le processus formation des détails D_0 à partir de S_0 est schématisé figure 3.5.

$$\begin{array}{ccccc} S_0 & & & & \\ \downarrow^a & \searrow & & & \\ S_1 & & U_0 & \rightarrow_c & D_0 \\ & \searrow_b & & \nearrow & \\ & & \tilde{S}_0 & & \end{array}$$

Figure 3.5. Extraction des détails D_0 par: a) sous-échantillonnage, b) prédiction, c) correction. Le signal sous-échantillonné constitue le signal S_1.

Partant d'un signal sous-échantillonné S_1, on peut alors reconstruire le signal S_0 par prédiction puis correction du signal prédit grâce aux détails D_0, comme schématisé figure 3.6.

$$\begin{array}{ccccc} S_1 & \rightarrow_a & \tilde{S}_0 & \rightarrow_b & S_0 \\ & & & \nearrow & \\ & & D_0 & & \end{array}$$

Figure 3.6. Reconstitution de S_0 à partir de S_1 et D_0 par a) prédiction et b) correction.

Le processus est alors itéré successivement sur le signal S_1 jusqu'à épuisement des échantillons. Lorsque le sous-échantillonnage n'est plus possible, l'ensemble S_n est alors choisi comme ensemble de détails D_n.

$$\begin{array}{ccccccccc} S_0 & \rightarrow & S_1 & \rightarrow & \dots & \rightarrow & S_n & \rightarrow & \{\} \\ & \searrow & & \searrow & & & & \searrow & \\ & & D_0 & & D_1 & & \dots. & & D_n = S_n \end{array}$$

Figure 3.7. Décompositions successives du signal S_0 en niveaux de détails D_i

La recomposition de S_0 s'effectue naturellement en corrigeant successivement les signaux prédits $\tilde{S}_i$ en fonction des détails D_{i+1}.

$$\begin{array}{ccccccccccc} \{\} & \rightarrow & S_n = D_n & \rightarrow & S_{n-1} & \rightarrow & \dots & \rightarrow & S_1 & \rightarrow & S_0 \\ & \nearrow & & \nearrow & & & & \nearrow & & \nearrow & \\ D_n & & D_{n-1} & & & & D_1 & & D_0 & & \end{array}$$

Figure 3.8. Recomposition du signal S_0 grâce aux niveaux de détails successifs D_i

L'ensemble du processus de décomposition/recompostion du signal est schématisé sur les figures 3.7 et 3.8. On notera toutefois que le *lifting scheme* original comporte une passe supplémentaire visant à garantir la conservation d'une certaine quantité (*e.g.* la valeur moyenne) lors du passage de S_i à S_{i+1}. Cependant cela nécéssite de stocker des données supplémentaires, aussi Lee & Shin suggèrent d'omettre cette partie pour une application de compression.

Dans le cas de la méthode de Lee & Shin pour un signal de quaternions, nous noterons s_i les éléments de S_0 et $\tilde{s}_i$ ceux de $\tilde{S}_0$. Le sous-échantillonnage consiste simplement à ne conserver que les éléments d'indices pairs *i.e.* les s_{2k}, ou $k \in K = \left[\!\left[0, \frac{\#S_0}{2} \right]\!\right]$. La prédiction $\tilde{s}_{2k+1} \in \tilde{S}_0$ de s_{2k+1} en fonction de s_{2k} et s_{2k+2} est donnée par l'interpolation sphérique à la valeur $\frac{1}{2}$ des s_{2k} et s_{2k+2}, soit :

$$\begin{aligned} \tilde{s}_{2k+1} &= \text{pred}(s_{2k}, s_{2k+1}) \\ &= \text{slerp}\left(s_{2k}, s_{2k+2}, \frac{1}{2} \right) \end{aligned} \tag{3.4}$$

La figure 3.9 montre une interprétation géométrique de la prédiction $\tilde{s}_{2k+1}$:

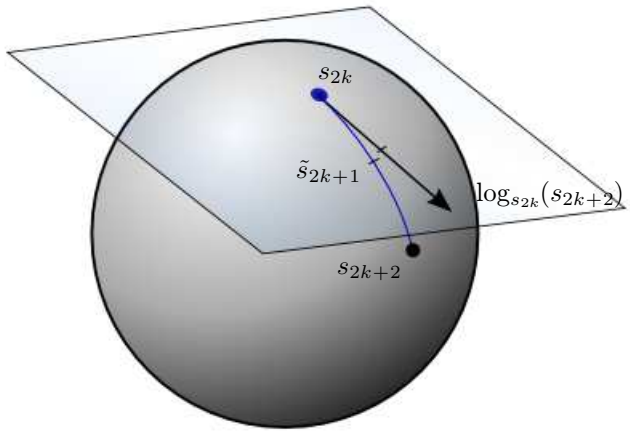


Figure 3.9. La prédiction $\tilde{s}_{2k+1}$ comme interpolation sphérique de s_{2k} et s_{2k+2}.

Pour corriger le signal prédit, nous formons le détail:

$$\begin{aligned} d_{2k+1} &= \log_{\tilde{s}_{2k+1}}(s_{2k+1}) \\ &= \ln\left(\tilde{s}_{2k+1}^{-1} . s_{2k+1} \right) \end{aligned} \tag{3.5}$$

Il s'agit de l'image de s_{2k+1} dans l'espace tangent à S^3 au point $\tilde{s}_{2k+1}$, $T_{\tilde{s}_{2k+1}}(S^3)$ comme présenté figure 3.10 :

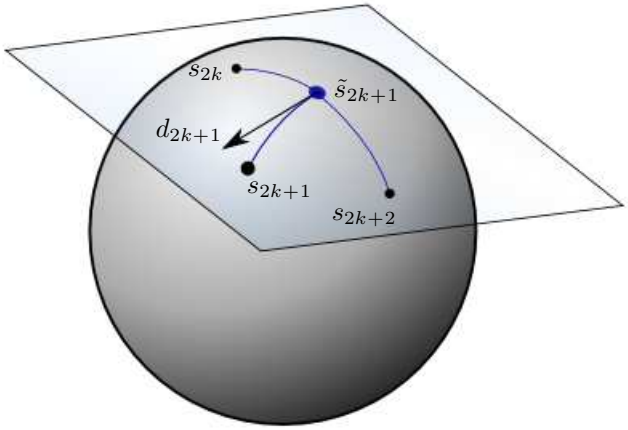


Figure 3.10. Le détail d_{2k+1} en fonction de s_{2k}, s_{2k+1} et s_{2k+2}.

L'intéret de la méthode en vue de la compression des données dépend fortement du pouvoir prédicteur de l'operation de reconstruction. Si la prédiction est suffisemment bonne, on pourra alors négliger les détails en dessous d'un certain seuil, ou même supprimer des niveaux entiers de résolution. Chaque retrait du dernier niveau de détail d'une animation divise par *deux* la taille des données. Nous avons choisi d'effectuer la compression d'animations en omettant certains niveaux de détails, notemment ceux relatifs aux plus hautes fréquences, qui constituent les gains les plus élevés au prix de la plus faible dégradation de qualité.

Le paramètre de compression est donc la donnée des niveaux de détails omis lors de la reconstruction.

Nos expériences avec cette méthode montrent que des taux de compression tout-à-fait honorables peuvent être atteints, tout en conservant l'allure globale de l'animation. Les taux de compression sont d'autant plus élevés que la durée de l'animation ainsi que sa fréquence d'échantillonnage sont importantes.

Le fait de travailler avec des données d'orientation cumule les erreurs le long de la hiérarchie, aussi les trajectoires des extrêmités du squelette sont parfois légèrement différentes de l'original. Ceci est particulièrement gênant en ce qui concerne les pieds puisque cela produit un effet de glissade des pieds *(footskate* en anglais), un artéfact particulièrement remarquable sur lequel nous reviendrons dans la partie 4.2.2.

3.4.2 Améliorations

Cette technique est généralisable aux variétés différentiables [RDS+05] même si son application pratique doit souvent se restreindre à certains groupes de Lie (dont SO(3)), pour lesquels le passage à l'espace tangent *via* la carte exponentielle est facilement calculable.

En se placant du point de vue de l'espace tangent au point s_{2k} (*cf.* les notations de la partie 3.4.1), la phase de prédiction par SLERP consiste en fait à interpoler linéairement le vecteur $\log_{s_{2k}}(s_{2k+2})$ à la valeur $\frac{1}{2}$, ce qui revient à se placer à mi-distance sur la géodésique joignant s_{2k} à s_{2k+2}. Ce faisant, on interpole ainsi linéairement les données dans les espaces tangents successifs (cf. figure 3.10).

Lors d'une reconstruction au cours de laquelle on ignore les details à partir d'un certain niveau, les données reconstruites seront ainsi "géodésiques par morceaux" comme présenté figure 3.11 :

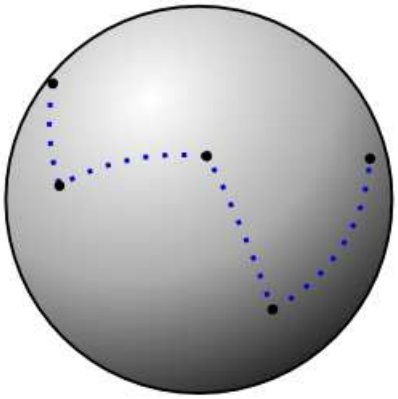


Figure 3.11. Allure du signal reconstruit en utilisant une prédiction par SLERP, en omettant les niveaux de détail les plus élevés.

Ce type de courbes présente des discontinuités de la dérivée première, ce qui se remarque lors de la lecture de l'animation reconstruite. L'animation est hachée et un aspect "robot", "linéaire par morceaux" émane du mouvement. La vitesse angulaire est de surcroît constante entre deux points, précisement comme c'est le cas lors d'une SLERP, ce qui accentue encore l'allure mécanique du mouvement reconstruit.

En nous inspirant des travaux de [RDS+05], nous avons donc implementé deux autres méthodes de prédiction.

La première consiste à ne rien prédire du tout, et s'apparente à une interpolation constante. Comme on s'y attend, les animations compressées par cette méthode présentent des discontinuités importantes et la méthode est donc inenvisageable en pratique.

La deuxième, développée dans [RDS+05] consiste à utiliser l'interpolation spline dans l'espace tangent. Lorsque c'est possible, *i.e.* lorsque l'on dispose d'au moins 4 points du signal: s_{-1}, s_0, s_1, s_2 aux instants $t = -1, 0, 1, 2$, on effectue l'interpolation spline dans l'espace tangent au point s_0 à l'instant $t = 0.5$ pour prédire le signal entre s_0 et s_1.

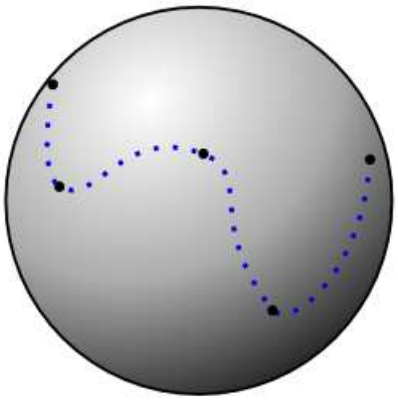


Figure 3.12. Allure du signal reconstruit en utilisant une prédiction par spline, en omettant les niveaux de détail les plus élevés.

L'avantage de cette méthode est qu'étant donné que l'application exponentielle est de classe $\mathcal{C}^\infty$, le signal reconstruit sera aussi lisse que la spline employée pour l'interpolation. Les animations compressées en utilisant cette technique ne souffrent plus des défauts évoqués pour le cas linéaire, et le mouvement semble cette fois tout-à fait naturel, et ce même dans le cas de fortes compressions.

Son principal désavantage est d'être parfois "trop lisse" dans le cas d'une reconstruction partielle, en ceci que les discontinuités de la dérivée première lors du contact des pieds avec le sol sont remplacées par de légères oscillations au voisinage de la zone de contact. Le fait d'omettre des niveaux de détails en vue de la compression résulte en un signal reconstruit lissé par les prédictions splines successives. S'il engendre des mouvements plus fluides et plus naturels, ce genre d'effets n'est cependant pas désirable dans tous les cas. Nous revenons sur ce point dans la partie 4.2.3.

La compression d'animations par élimination des niveaux de détail les plus élevés permet d'obtenir des résultats tout-à-fait satisfaisants, pour des taux de compression relativement élevés (souvent supérieurs à 30:1, sans pour autant quantifier les données sur 16 bits comme c'est le cas dans les autres travaux [BPvdP07],[Ari06]. Quelques résultats de cette méthode sont présentés en partie 6.

Nous avons cependant cherché à améliorer encore les taux obtenus en compressant les niveaux de détail eux-mêmes. Nous présentons à présent une méthode statistique de compression de données d'orientations, puis l'adaptons à l'analyse multi-résolution présentée ci-dessus.

3.5 Analyse statistique de données d'orientation

La technique évoquée au paragraphe précedent exploite principalement la cohérence temporelle des données pour parvenir à compresser le signal. Cependant elle néglige totalement les corrélation et les redondances pouvant exister dans le mouvement des différentes articulations du squelette.

Pourtant, lors d'un mouvement de locomotion tel que la marche, les données ne présentent manifestement pas tous les degrés de liberté dont fait preuve le squelette : les bras évoluent le long du corps, les jambes restent à la verticale, et le côte cyclique de la marche laisse présager une grande redondance dans la distribution des données au cours du temps.

Pollard et al. [SHP04] utilisent par exemple une analyse en composantes principales sur les positions des articulations afin de déceler les corrélations dans les mouvements, et de réduire la dimension de l'espace du mouvement afin de faciliter les techniques d'optimisation.

Une analyse statistique des données d'orientation permettrait donc, comme c'est le cas pour les positions, de déceler d'éventuelles corrélations dans les mouvements articulaires, ceci afin de reduire le nombre de degres de libertés du mouvement, et ainsi permettre de la compression.

Le problème majeur, encore une fois, est que les outils de statistique dans des espaces non-linéaires sont plus rares et plus délicats à manier que leurs homologues linéaires. On trouve néanmoins des travaux récents sur le sujet, tels l'analyse en géodésiques principales de Fletcher [FLPJ04] dont on décrit brièvement le fonctionnement ci-dessous.

3.5.1 Moyenne de Karcher

Dans le cadre général d'une variété riemannienne[3.7], il est impossible de définir la valeur moyenne de données comme espérance probabiliste d'une variable aléatoire, car l'espérance est par définition une valeur de $\mathbb{R}$ (ou de $\mathbb{R}^n$), car reliée à la *mesure* d'une partie. Cependant, on remarque que la valeur moyenne dans le cas Euclidien minimise les distances aux données : [Moa02]

$$\mu \;=\; \arg\min_{x\in\mathbb{R}^n} \sum_{i\in I} \|x_i - x\|^2$$

Dans le cas Euclidien, l'existence et l'unicité de la solution de ce problème d'optimisation est garantie.

La métrique riemanienne induit sur la variété une distance, qui est l'intégration des normes des vecteurs tangents à la géodésique joignant deux points. Pour garantir l'existence et l'unicité d'une géodésique de longueur minimale entre deux points, on se placera dans le cadre des variétés *complètes* géodésiquement. La distance s'exprime alors en termes de norme Euclidienne dans l'espace tangent, auquel on passe par la fonction logarithme :

$$\forall x, y \in M, d(x, y) = \|\log_x(y)\|$$

Cette distance permet de définir la valeur moyenne de données dans une variété riemannienne comme solution du problème de minimisation suivant :

3.7. Une variété riemannienne M est une variété différentiable pour laquelle l'espace tangent en tout point $p \in M$ est muni d'un produit scalaire variant continuement avec p le long de la variété. Cela permet entre autres de définir un élément de longueur infinitésimal, puis par intégration de ceux-ci de définir la longueur de courbes. La longueur d'une géodésique entre deux points permet alors définir la distance entre les points.

Définition 3.2. *Moyenne de Karcher [Kar77]*

Soit $I = [\![1, N]\!]$ et $(x_i)_{i\in I} \in M$ un ensemble de N points d'une variété riemannienne M. Sous réserve d'existence et d'unicité de la solution au problème de minimisation ci-dessous, on définit la moyenne de Karcher μ des $(x_i)_{i\in I}$ par:

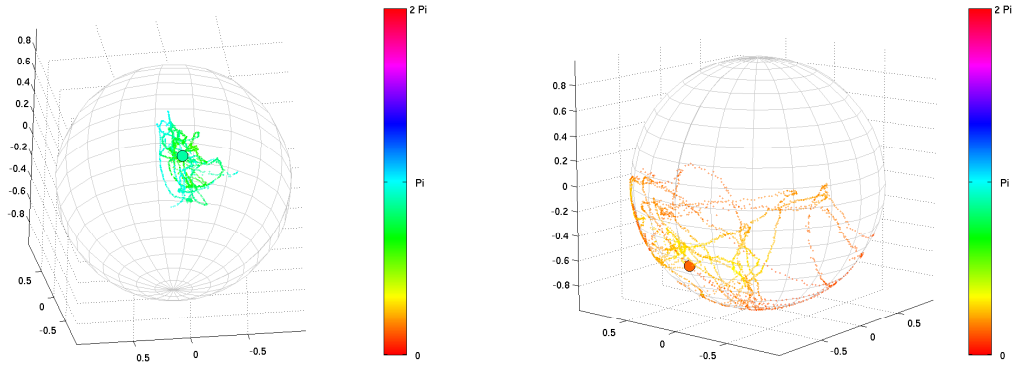
$$\begin{aligned} \mu &= \underset{i\in I}{\text{Karcher}}(x_i) \\ &= \arg\min_{x\in M} \sum_{i\in I} d(x, x_i)^2 \end{aligned}$$

L'existence ainsi que l'unicité de la solution ne sont pas garanties dans le cas général sauf pour des données bien localisées [Ken90], mais Fillmore les a démontrées [BF01] dans le cas particulier des sphères $S^n, n \geqslant 1$ sans restriction sur les données, et donne un algorithme itératif de type descente de gradient pour le calculer :

Algorithme 1. Moyenne de Karcher

Entrée: $(x_i)_{1\leqslant i\leqslant N} \in S^n$
Sortie : $\mu \in M$, la moyenne de Karcher des x_i
$\mu_0 = x_1$
Faire
$\Delta\mu = \frac{1}{N}\sum_{i=1}^{N} \log_{\mu_j}(x_i)$
$\mu_{j+1} = \exp_{\mu_j}(\Delta\mu)$
Tantque $\|\Delta\mu\| > \varepsilon$

La figure 3.13 montre les résultats obtenus par l'algorithme pour des données de capture de mouvement.



a) 12 itérations b) 10 itérations

Figure 3.13. Les résultats de l'algorithme pour les données d'orientation d'un humérus gauche (a), et d'un fémur droit (b) pour une animation de danse. La moyenne de Karcher est representée par le cercle. Le nombre d'itérations de l'algorithme est indiqué dans chaque cas.

L'orientation sur la sphère représente l'axe d'un quaternion, et la couleur son angle.

On notera qu'en fin d'algorithme, les données dans l'espace tangent à la moyenne de Karcher sont de moyenne euclidienne nulle (c'est la condition de sortie de l'algorithme). L'algorithme étant de type descente de gradient d'ordre de convergence linéaire, une trentaire d'itérations suffisent largement à approcher la précision de la machine. [Pen99] indique qu'entre 5 et 10 sont généralement suffisantes, avec toutefois de bonnes conditions initiales.

3.5.2 Analyse en géodésiques principales (AGP)

Fletcher propose un analogue de l'analyse en composantes principales pour les données d'une variété Riemannienne, en projetant les données sur les géodésiques maximisant la variance des données. Pour ce faire, il définit le projeté d'un point sur une géodésique H comme le point de la géodésique minimisant la distance géodésique au point considéré :

$$\forall x \in M, \pi_H(x) = \arg\min_{y \in H} d(x, y)^2$$

Exception faite des cas dégénérés assez rares, cette projection est bien définie. En approximant cet operateur de projection par la projection orthogonale dans l'espace tangent à la moyenne des données, Fletcher propose alors une approximation linéaire de l'AGP se réduisant à une ACP traditionnelle des données dans l'espace tangent à la moyenne. Les vecteurs propres de l'ACP correspondent aux géodésiques passant par la moyenne décrivant le mieux la variabilité des données. Cette approximation est en quelque sorte la "meilleure" approximation linéaire possible. L'algorithme est donnée de manière synthétique ci-dessous.

Algorithme 2. Analyse en Géodésiques Principales

Entrée: $(x_i)_{1 \leqslant i \leqslant N} \in M$
Sortie: Directions principales, $v_k \in T_\mu(M)$; Variances selon les directions v_k, $\lambda_k \in \mathbb{R}$
$\mu = \text{Karcher}(x_i)$
$u_i = \log_\mu(x_i)$
$S = \frac{1}{N} \sum_{i=1}^{N} u_i . u_i^T$
$\{v_k, \lambda_k\} = \text{vecteurs/valeurs propres de } S$

Dans le cas de données de S^3, l'ACP des logarithmes de quaternions donne 3 vecteurs propres orthogonaux constituant une base de $T_\mu(S^3) = \mathbb{R}^3$. Les 3 directions associées correspondent à 3 géodésiques passant par la moyenne des données. La figure 3.14 montre des résultat de l'AGP pour le mouvement d'articulations dans une animation issue de capture de mouvement.

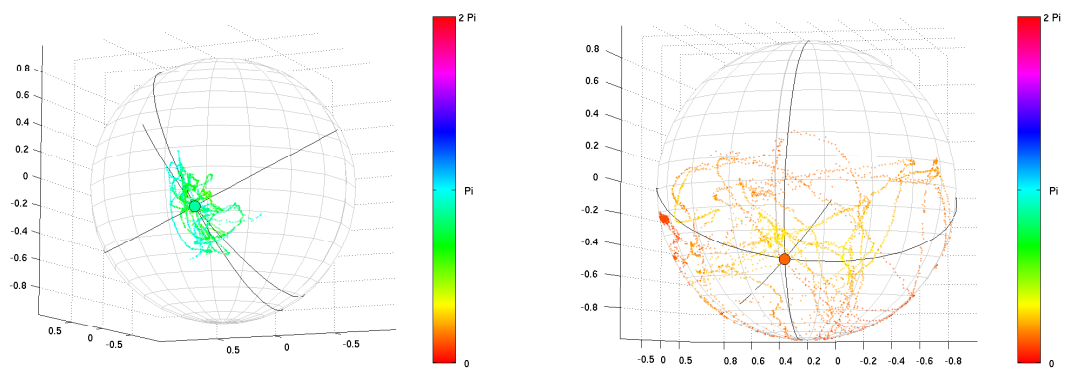


Figure 3.14. Le résultat de l'AGP sur les données d'orientation présentées figure 3.13. Les géodésiques issues de l'AGP sont ici indiquées en noir.

Dans le but de prendre en compte les corrélations entre les différents mouvements angulaires, nous avons étendu la méthode d'approximation au cas de plusieurs signaux de quaternions.

Nous représentons une animation squelettique par une matrice $(q_{i,j})$ de quaternions de taille $N \times M$, où N désigne le nombre d'échantillons de l'animation, et M désigne le nombre de joints du squelette. La $i^{\text{ème}}$ ligne de cette matrice définit la *pose* à l'instant i, *i.e* la configuration angulaire de chacune des articulations à l'instant i. La $j^{\text{ème}}$ colonne définit quant à elle le mouvement angulaire du $j^{\text{ème}}$ joint du squelette au cours de l'animation.

De manière à prendre en compte les corrélations existant entre les mouvements des différentes articulations lors de l'ACP, nous effectuons l'ACP sur une matrice réunissant les logarithmes de quaternions des différents joints centrés en leurs moyennes respectives, comme nous le ferions dans le cas linéaire pour prendre en compte les corrélations existant entre plusieurs variables vectorielles. Pour ce faire, nous formons le vecteur des moyennes pour chaque joint :

$$\boldsymbol{\mu} = (\mu_j)_{1 \leqslant j \leqslant M}$$

Où chaque μ_j est défini comme moyenne de Karcher des quaternions du $j^{\text{ème}}$ joint :

$$\forall j \in [\![1, M]\!], \mu_j = \underset{1 \leqslant i \leqslant N}{\text{Karcher}}(q_{i,j})$$

Nous formons alors une matrice $D = (d_{i,j})$ de taille $N \times M$ dont chaque colonne contient les logarithmes en μ_j des quaternions $(q_{i,j})_{1 \leqslant i \leqslant N}$:

$$\forall i \in [\![1, M]\!], \forall i \in [\![1, N]\!], d_{i,j} = \log_{\mu_j}(q_{i,j}) \tag{3.6}$$

Nous effectuons l'ACP sur cette matrice D. Les vecteurs propres engendrés par l'ACP sont alors la concaténation de j vecteurs des différents espaces tangents aux moyennes respectives. Ces j vecteurs décrivent au mieux la variabilité du mouvement de l'*ensemble* des joints, et correspondent chacun à une géodésique de S^3 passant par μ_j.

3.5.3 Compression par AGP de données réelles

Nous avons appliqué cette méthode aux données d'orientation issues de la motion capture afin de mettre en évidence des corrélations pouvant être exploitées en vue de la compression. Pour ce faire, nous calculons la moyenne de Karcher des données d'orientation pour chaque joint par l'algorithme 1.

Nous formons par la suite la matrice D de l'équation 3.6, et lui appliquons l'analyse en composantes principales. Il apparaît que pour des données angulaires présentant 90 degrés de liberté, une dizaine de vecteurs propres seulement suffisent pour représenter 98% et de la variance des données, comme l'indiquent les graphes de variance cumulée ci-dessous. Nous pouvons alors omettre les vecteurs propres restants, ainsi que les coordonnées des détails sur ces vecteurs, afin de compresser les données comme dans le cas d'une compression par ACP classique. Ce processus est décrit figure 3.15.

$$\begin{array}{ccccccc} & & & & (v_k)_{k \leqslant M} & \rightarrow & (v_k)_{k \leqslant P} \\ D \in \mathcal{M}_{N \times M}(\mathbb{R}) & \rightarrow & \text{ACP} & \nearrow & & & \\ & & & \searrow & D' \in \mathcal{M}_{N \times M}(\mathbb{R}) & \rightarrow & D'' \in \mathcal{M}_{N \times P}(\mathbb{R}) \end{array}$$

Figure 3.15. La compression par ACP de la matrice D: l'ACP de D fournit M vecteurs propres (v_k), et les composantes D' sur ces M vecteurs. En ne gardant que les P premiers vecteurs propres, nous pouvons garder uniquement les P premières colonnes de D', d'ou un taux de compression de M/P.

Les graphes des figures 3.16, 3.17 et 3.18 présentent la somme cumulée des k premières valeurs propres de la matrice de covariance, normalisée à 1. Nous constatons que les taux de compression obtenus varient entre 4 et 10 en ne gardant que les vecteurs propres représentant 98% de la variance des données d'entrée, ce qui correspond à une douzaine de vecteurs propres conservés en moyenne. Le taux de variance conservée constitue le paramètre de compression par AGP d'une animation.

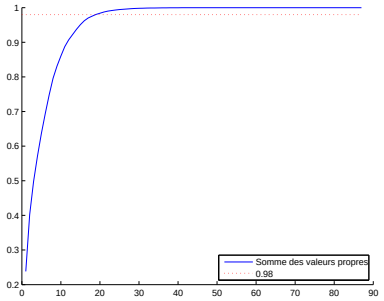


Figure 3.16. Animation de danse (4,71:1)

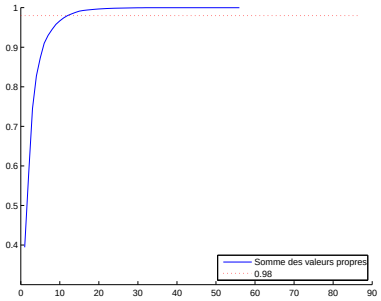


Figure 3.17. Animation de douche (6,68:1)

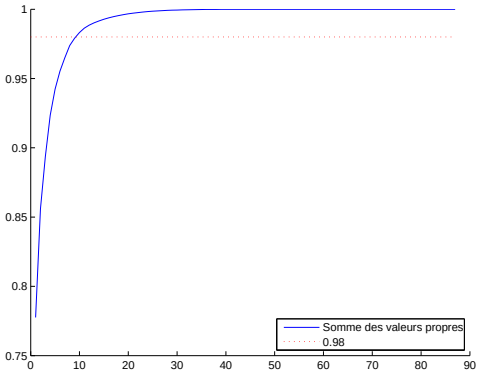


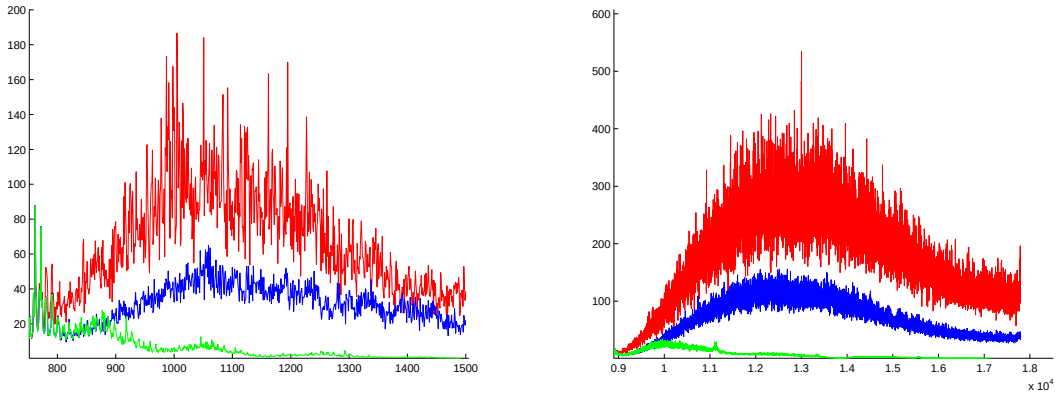
Figure 3.18. Animation de marche. (8,46:1)

Nous remarquons également que les mouvements présentant de fortes redondances tels que la marche sont les mieux compressés par cette méthode.

Malheureusement, la reconstruction de l'animation pose problème : si l'allure générale de l'animation reconstruite ressemble beaucoup à l'originale, des hautes fréquences parasitent le mouvement et pénalisent la qualité visuelle du résultat, à plus forte raison aux extrêmités du squelette.

La figure 3.19 présente une comparaison des spectres de Fourier pour les données de positions des joints du squelettes entre l'animation originale et celle reconstruite. Dans un souci de lisibilité, un filtre passe-haut[3.8] a été appliqué aux deux animations. Il apparaît clairement que le signal reconstruit présente beaucoup plus de hautes fréquences que le signal original.

3.8. Le filtre en question est une différence du signal et de sa convolution avec un noyau gaussien



a) animation de marche b) animation de danse

Figure 3.19. Comparaison des spectres de Fourier des positions entre animation originale (en bleu), compressée par AGP (en rouge) et par multi-résolution en omettant les deux derniers niveaux de détails (vert). Les positions ont subi un filtre passe-haut. L'animation compressée par AGP présente clairement beaucoup plus de hautes fréquences que l'originale. Les propriétés lissantes de la compression multi-résolution apparaissent également de manière flagrante.

Nous avons alors logiquement pensé à utiliser les propriétés de filtrage de la représentation multi-résolution du paragraphe précédent en combinaison avec l'analyse en géodésiques principales afin d'améliorer les taux de compression fournis par la première méthode tout en évitant les hautes fréquences parasites présentes dans la seconde.

3.6 Une méthode hybride

Nous avons appliqué l'analyse en géodésiques principales afin d'obtenir une représentation plus compacte des différents niveaux de détails fournis par l'analyse multi-résolution décrite au paragraphe 3.4.1. S'il n'est pas évident à première vue que la représentation par niveaux de détails présente des corrélations du même type que le signal original, elle peut tout du moins mettre en évidence des redondances statistiques exploitables à des fins de compression.

Afin d'effectuer l'AGP sur les différents niveaux de détails, nous avons adapté l'algorithme de réduction du signal en base+détails. En reprenant les notations de la partie 3.4.1, nous formons toujours les quaternions de correction : $c_{2k+1} = \tilde{s}_{2k+1}^{-1}.s_{2k+1}$ de la prédiction $\tilde{s}_{2k+1} = \text{pred}(s_{2k}, s_{2k+2})$, de manière à reconstruire le signal original de la manière suivante :

$$\begin{aligned} \text{pred}\big(s_{2k}, s_{2k+2}\big).c_{2k+1} &= \tilde{s}_{2k+1}.\big(\tilde{s}_{2k+1}^{-1}.s_{2k+1}\big) \\ &= \big(\tilde{s}_{2k+1}^{-1}.\tilde{s}_{2k+1}\big).s_{2k+1} \\ &= s_{2k+1} \end{aligned}$$

par l'associativité de la multiplication des quaternions.

Toutefois, nous ne stockons plus directement le détail $d_{2k+1} = \log(c_{2k+1})$. Nous effectuons à la place une AGP de ces quaternions de correction, afin de réduire l'espace de stockage de chaque niveau de détail. À cet effet, nous calculons la moyenne de Karcher μ des c_{2k+1} par l'algorithme 1 :

$$\mu = \underset{k \in K}{\text{Karcher}} \ \{c_{2k+1}\}$$

Puis nous formons les images d_{2k+1} des quaternions de correction pour le niveau courant dans $T_\mu(S^3)$, l'espace tangent à la moyenne μ :

$$\begin{aligned} d_{2k+1} &= \log_\mu(c_{2k+1}) \\ &= \log\big(\mu^{-1}.c_{2k+1}\big) \end{aligned}$$

Ceci nous permet d'approximer l'AGP des corrections c_{2k+1} par une ACP traditionnelle des d_{2k+1}, d'après la méthode proposée par Fletcher. La compression s'effectue alors en ne gardant que les vecteurs propres de la matrice de covariance contribuant le plus à la variabilité des données, ainsi que les coordonnées des d_{2k+1} sur ces vecteurs propres, comme dans le cas d'une compression par ACP classique.

Nous sommes alors en mesure de reconstruire approximativement les d_{2k+1} grâce aux vecteurs propres et aux coordonnées sur ceux-ci, puis de reconstruire les quaternions de correction:

$$\begin{aligned} \mu.\exp(d_{2k+1}) &= \mu.\exp\big(\log\big(\mu^{-1}.c_{2k+1}\big)\big) \\ &= \mu.\mu^{-1}.c_{2k+1} \\ &= c_{2k+1} \end{aligned}$$

Nous pouvons alors reconstruire une approximation du signal original en corrigeant le signal prédit $\tilde{s}_{2k+1}$ par les c_{2k+1}.

Pour chaque niveau de détails, nous devons donc stocker :

1. La moyenne de Karcher des détails, que l'on peut elle-même stocker sous forme de logarithme de quaternion.

2. Les vecteurs propres issus de l'ACP des détails d_{2k+1}, éventuellement en ne gardant qu'un sous-ensemble de ceux-ci représenant le mieux les données.
3. Les coordonnées des d_{2k+1} dans la base (partielle) de vecteurs propres choisis en 2).

Dans le but d'obtenir des niveaux de détails de taille suffisemment imporante pour que l'AGP ait du sens, nous stoppons la décompostition en niveaux de détails lorsque le signal sous-échantillonné possède moins d'échantillons que le nombre de joints du squelette. Il serait en effet contre-productif pour une application à la compression d'effectuer des AGP de piètre qualité tout en stockant des vecteurs propres additionnels.

En corrolaire de ceci, une animation pour laquelle $N < M$ ne pourra pas être traitée par notre méthode hybride, et l'on se rabattra alors sur la version multi-résolution simple dans le but d'éviter les hautes fréquences parasites (cf. partie 3.5.3). Fort heureusement, les animations de ce type correspondent à des durées géneralement très courtes (de l'ordre de la seconde, selon la fréquence d'échantillonnage) et sont assez rares en pratique.

Dans le cas contraire, la méthode s'applique et les résultats obtenus sont présentés dans la partie 6. Les taux de compression obtenus sont souvent bons voire très bons, cependant les animations résultantes nécéssitent un traitement post-opératoire dans le but de corriger quelques artéfacts visuels tels que le glissement des pieds au sol. Une discussion de la méthode est effectuée à la partie 6.

D'après nos expériences, l'AGP sur chaque niveau de détail de la version multi-résolution donne en général une vingtaine de vecteurs propres par niveau de détails pour conserver 98% de la variance des données, contrairement à une dizaine seulement dans le cas de la compression par AGP sur le signal original.

Ce résultat est assez surprenant, et nous pouvons conjecturer deux hypothèses à ce sujet :

- L'analyse multi-résolution décorrèle les données du signal original
- Dans le cas contraire, la *représentation* choisie pour les détails ne permet pas de faire apparaître ces corrélations lors de l'AGP

Quoi qu'il en soit, l'analyse en géodédiques principales permet de représenter la variabilité des données de manière plus compacte puisqu'elle autorise un gain allant de 3 à 5 par rapport aux données obtenues par l'analyse multi-résolution seule.

On remarquera cependant qu'il faut dans ce cas stocker les vecteurs propres de l'ACP, ce qui pénalise le taux de compression obtenu. Neanmoins le gain permis par l'AGP reste supérieur à la surcharge occasionnée, à plus forte raison pour les animations de longue durée.

Étant donné que l'AGP donne de moins bons résultats sur des niveaux de détails que sur un signal mono-résolution (éventuellement une version sous-échantillonnée du signal original), nous avons tout intérêt à reconstruire les premiers niveaux de détail du signal pour en obtenir une version sous-échantillonnée avant d'y appliquer l'AGP.

En effet, un signal et sa version multi-résolution contiennent le même nombre de données, cependant l'AGP sur les niveaux de détails nécéssite de stocker une base de vecteurs propres par niveau de détail, pour des résultats globalement moins bons comme évoqué ci-dessus. Nous pourrons alors utiliser les niveaux de détail restants pour poursuivre la reconstruction et ainsi bénéficier du lissage offert par les ondelettes splines.

Ce faisant, on veillera néanmoins à ne pas reconstruire entièrement le signal et à conserver quelques niveaux de détails (même vides) afin que les propriétés de lissage de la reconstruction s'appliquent. Dans le cas contraire nous serions ramenés au cas de la partie 3.5.3 et le résultat souffrirait de hautes fréquences parasites. D'après nos expériences, les trois niveaux de détails les plus fins permettent généralement de lisser le bruit occasionné par l'AGP.

Notons enfin que dans l'implémentation, les données de position de l'articulation racine ont également été compressées par analyse multi-résolution simple en utilisant une reconstruction par prédiction spline. Les niveaux de détails conservés sont alors les mêmes que pour les données d'orientation.

3.7 Implémentation

L'implémentation de la méthode de compression présentée s'est faite grâce au logiciel MATLAB[3.9], avec l'aide des boîtes à outils (*toolboxes*) MOCAP[3.10] et QTFM[3.11]. Les données utilisées proviennent principalement de la base de données d'animations de motion capture en libre consultation de Carnegie Mellon University[3.12].

L'application permet à l'utlisateur d'effectuer la compression et la visualisation d'animations au format BVH[3.13], puis de sauvegarder le résultat pour une visualisation plus agréable dans le logiciel Maya[3.14], ceci afin d'exporter des vidéos. La figure 3.20 propose une capture d'écran de l'application en fonctionnement.

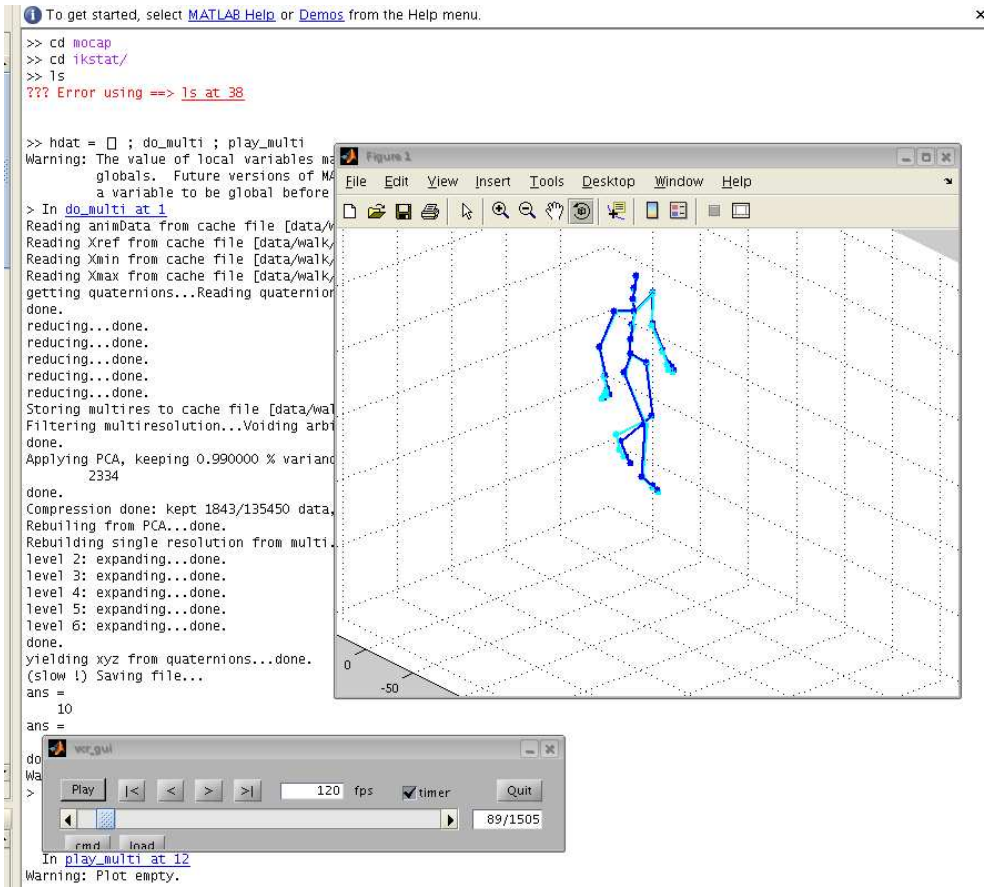


Figure 3.20. Un capture d'écran de l'implémentation en MATLAB de la méthode de compression présentée. Le composant d'affichage d'animations en capture de mouvement est issu de la boîte à outils MOCAP.

3.9. MathWorks MATLAB, http://www.mathworks.com/

3.10. MATLAB Motion Capture Toolbox, http://www.cs.man.ac.uk/~neill/mocap/

3.11. Quaternion Toolbox For MATLAB, http://qtfm.sourceforge.net/

3.12. CMU Graphics Lab Motion Capture Database, http://mocap.cs.cmu.edu/

3.13. BioVision BVH. Une description du format de fichier est disponible à l'adresse http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html

3.14. Autodesk Maya, http://www.autodesk.com/maya

La machine de travail était un Intel Core2 Duo équippé de 1.5 Go de mémoire vive, fonctionnant sous le système d'exploitation libre GNU/Linux.

Chapitre 4
Bilan

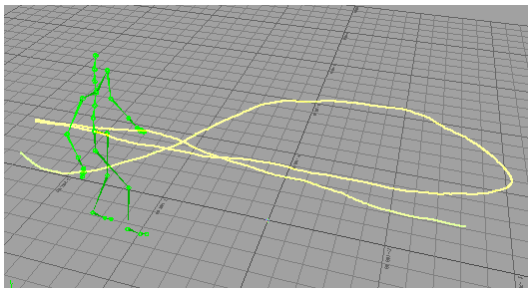
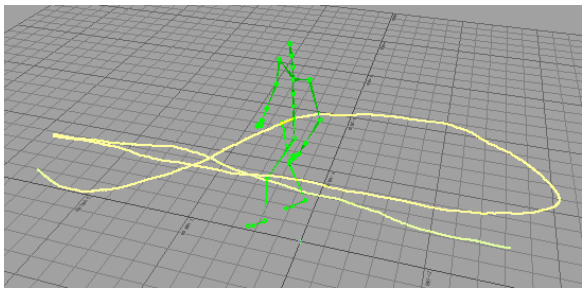
4.1 Résultats

Nous présentons maintenant les résultats obtenus avec notre méthode de compression. Nous proposons divers exemples d'animations compressées représentant divers types de mouvement de longueurs différentes, avec pour chacune :

- Les différents paramètres de compression:
  - Le nombre de niveaux de détails de la représentation multi-résolution conservés
  - La variance conservée lors de l'AGP des détails
- Le taux de compression obtenu
- Une estimation du temps de calcul
- Le nombre de vecteurs propres obtenus lors de l'AGP
- Les graphes d'erreur en norme quadratique ainsi que géodésique de l'animation reconstruite
- Le type d'ondelettes utilisées lors de l'analyse multi-résolution
- Des captures d'écran comparant l'animation originale à sa version compressée, avec les trajectoires de certains membres
- Une discussion de l'efficacité de la méthode dans le cas de l'exemple proposé

Nous poursuivons par une discussion d'ordre plus général sur la méthode de compression proposée, et proposons quelques pistes de réflexion en vue de futurs travaux.

4.1.1 Animation de marche

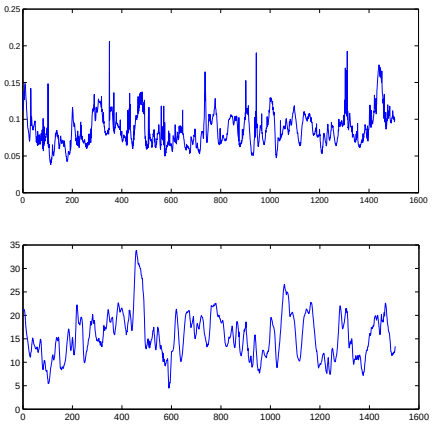



Cette animation représente une marche lente aléatoire dans différentes parties de la scène. Son analyse multi-résolution engendre 12 niveaux de détails, mais seuls les 6 derniers seront utilisés à des fins de compression. Les 6 premiers sont reconstruits pour former le signal à la plus basse résolution (*cf.* fin de la partie 3.6). Ce premier "niveau" constitue simplement un sous-échantillonnage de l'animation originale, aussi l'AGP sur ce niveau donne-t-elle de meilleurs résultats que sur les niveaux suivants, comme évoqué à la fin de la partie 3.6. Les temps de calculs pour la compression/decompression de cette animation sont de l'ordre de la minute.

4.1.1.1 Compression faible

Nous omettons le dernier niveau de détail, et appliquons l'AGP sur les niveaux restants en conservant 99% de la variance de chaque niveau. Nous regroupons en outre les 3 premiers niveaux en un seul afin d'améliorer la compression par AGP. Le dernier niveau étant omis, son AGP n'engendre aucun vecteur propre. Le taux de compression final obtenu est de 6.28:1.

Niveau de détail	# vecteurs propres
1	13
2	24
3	24
4	0



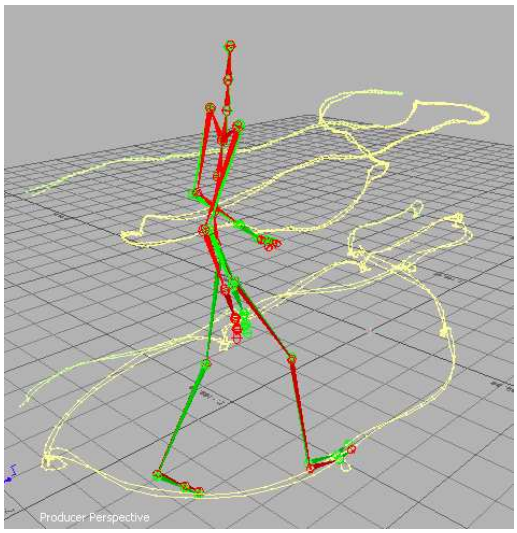


Tableau 4.1. Résultats obtenus pour une compression faible (6,28:1) d'une animation de marche.

Les graphiques présentent les vecteurs propres de l'AGP (*haut-gauche*) par niveau de détail, l'erreur en norme géodésique cumulée des orientations reconstruites(*centre-gauche*), et l'erreur en norme quadratique cumulée des positions(*bas-gauche*).

À droite, une capture d'écran comparant l'animation originale (*vert*) et l'animation reconstruite (*rouge*), ainsi que les trajectoires de la main droite et du pied gauche pour chacune.

L'aspect visuel de l'animation reconstruite est très proche de l'originale, sauf pour les pieds qui ont parfois tendance à glisser légerement sur le sol. Les différences de trajectoires sont plus marquées lors des changements de direction lors desquels les pieds glissent parfois franchement. L'erreur commise en norme géodésique reste relativement faible, mais se résulte en certains écarts importants sur les positions.

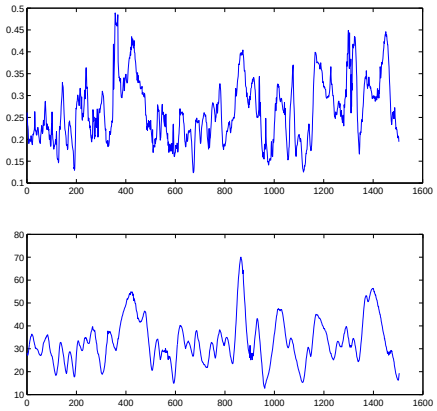
Enfin, nous notons que l'erreur oscille entre valeurs basses et hautes, ce qui est dû à l'analyse multi-résolution. En effet le signal reconstruit grâce à celle-ci interpole le signal original sous-échantillonné au niveau de détail le plus faible, aussi l'erreur dans le cas d'une compression sans AGP oscille-t-elle de la même manière avec cependant des minimum à 0 (les positions clés du signal sous-échantillonné sont reconstruites exactement). Nous remarquons donc que l'AGP introduit une erreur supplémentaire sur les orientations reconstruites (le minimum est cette fois de 0.05), qui se traduit par une erreur plus importante sur les positions. En outre, les positions clés ne sont plus reconstruites exactement.

Cependant, l'animation reconstruite est très convaincante visuellement, exception faite du *footskating*. Nous proposons des pistes pour son élimination dans la partie 6.

4.1.1.2 Compression forte

Nous omettons cette fois autant de niveaux de détails que possible pour permettre l'AGP, qui se fera donc uniquement sur une version sous-échantillonnée de l'animation. Nous ne gardons en outre que 90% de la variance ce celle-ci lors de l'AGP.

Niveau de détail	# vecteurs propres
1	4
2	0
3	0
4	0
5	0
6	0



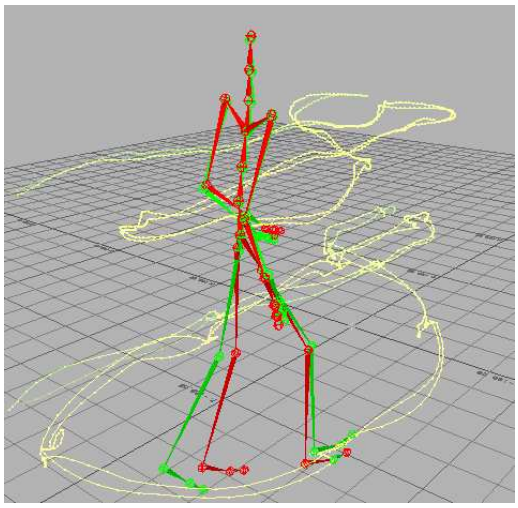


Tableau 4.2. Résultats obtenus pour une compression forte (215:1) d'une animation de marche. On constate que l'écart entre les trajectoires est cette fois significatif.

Le mouvement ressemble à l'original, mais le lissage est tellement important que l'animation a complètement perdu son style. Les positions des jambes sont de surcroît souvent incorrectes.

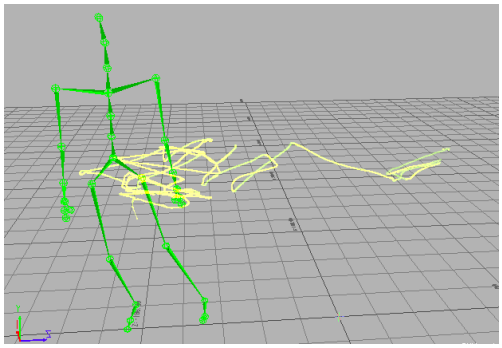
Il apparaît que l'AGP sur le premier niveau de détail pénalise l'*ensemble* de la reconstruction. En effet, même si d'un point de vue de compression ses résultats sont meilleurs, un premier niveau de détail approximatif se ressentira sur le résultat final puisque les niveaux de détail supérieurs fournissent des informations pour l'*interpolation* de ce premier niveau.

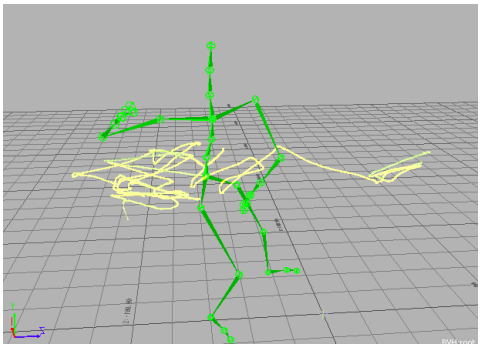
Nous avons donc deux solutions pour améliorer la qualité du résultat :

- Conserver plus de variance des données
- Ne pas effectuer l'AGP sur le premier niveau de détail

Nous montrons ci-dessous deux exemples mettant en place ces solutions, suivi d'un troisième les mélangeant.

4.1.2 Animation de danse



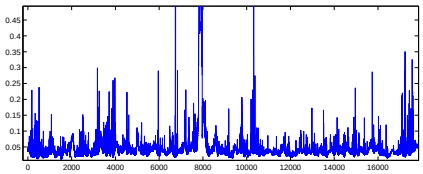


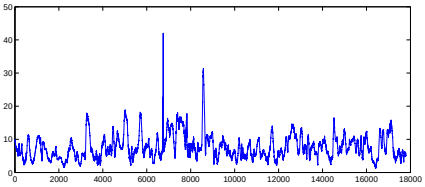
Cette animation est celle d'un danseur effectuant une chorégraphie. Ce mouvement est relativement long, et son analyse multi-résolution engendre 16 niveaux de détail. Nous n'utiliserons que les 10 derniers afin de compresser. Les temps de calculs pour la compression/décompression de l'animation sont de l'ordre de cinq minutes.

4.1.2.1 Compression faible

Nous éliminons les deux derniers niveaux de détail, et nous gardons désormais 99.9% de la variance des données. Nous reconstruisons les 4 premiers niveaux avant d'y appliquer l'AGP. Le taux de compression obtenu est de 8.27:1.

Niveau de détail	# vecteurs propres
1	33
2	41
3	40
4	40
5	40
6	0
7	0





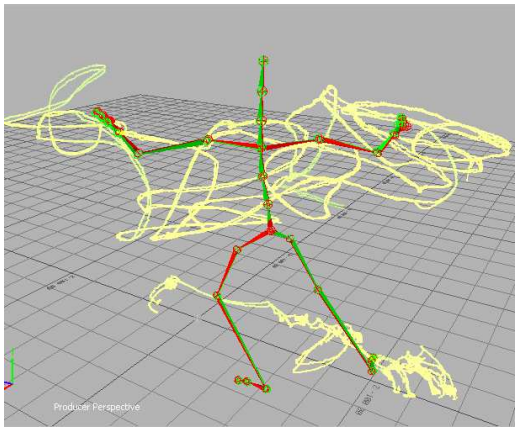


Tableau 4.3. Résultats obtenus pour une compression faible (8.27:1) d'une animation de danse. Les pics sur le graphe d'erreur géodésique (*centre-gauche*) sont dus au bruitage des données d'entrée : le poignet gauche y effectue une rotation complète. Cette rotation est atténué dans la version compressée, d'où l'écart important.

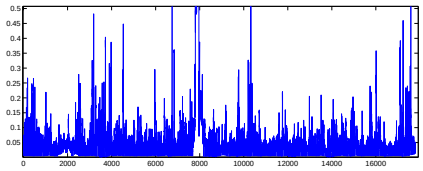
Les trajectoires originales et compressées sont dans ce cas très proches.

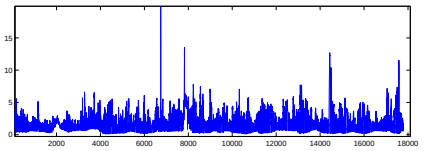
La qualité générale de cette animation reconstruite est excellente : elle "colle" littéralement à l'originale. Jouées côte-à-côte, il est difficile de les distinguer.

4.1.2.2 Compression forte

Nous éliminons cette fois 5 niveaux de détail, et gardons 95% de la variance. Nous reconstruisons les 4 premiers niveaux de détail, mais nous n'appliquons cependant pas l'AGP à ce premier niveau reconstruit. Le taux de compression est alors de 52,89:1.

Niveau de détail	# vecteurs propres
1	×
2	16
3	0
4	0
5	0
6	0
7	0





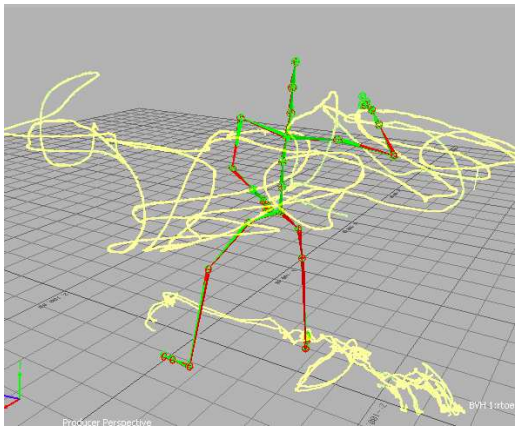


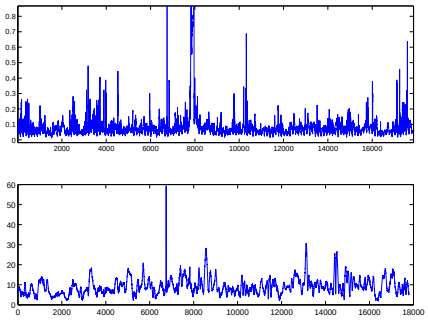
Tableau 4.4. Résultats obtenus pour une compression forte (52,89:1) d'une animation de danse. Les erreurs sont cette fois beaucoup plus faibles que dans tous les cas précédents. Les trajectoires sont pratiquement confondues.

Les résultats obtenus sont très bons : l'animation reconstruite est pratiquement la même que l'originale, avec de surcroît un leger lissage des données originales bruitées. Le taux de compression en lui-même est très encourageant. Nous présentons ci-dessous une compression encore plus forte.

4.1.2.3 Compression très forte

Nous éliminons cette fois les 5 niveaux de détails les plus fins. Nous reconstruisons les 3 premiers niveaux de détail, puis appliquons l'AGP à ce niveau en conservant 99.9% de la variance. Le taux de compression obtenu est alors de 108,92:1.

Niveau de détail	# vecteurs propres
1	33
2	13
3	11
4	0
5	0
6	0
7	0
8	0



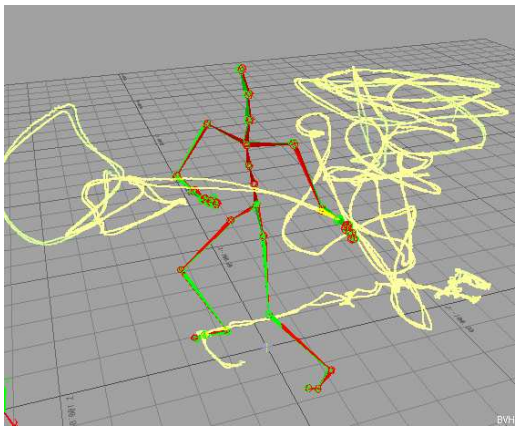


Tableau 4.5. Résultats obtenus pour une compression très forte (108,92:1) d'une animation de danse. Les erreurs ne sont pas beaucoup plus importantes que dans le cas précédent.

L'animation reconstruite est encore une fois très proche de l'originale, et ce même pour des taux de compression très élevés. Nous pensons que des taux de compressions encore plus élevés peuvent être atteints dans certains cas en ajustant les paramètres de compression.

4.2 Discussion - Perspectives

Nous avons présenté dans ce document une méthode de compression d'animations utilisant les logarithmes de quaternions. Ceux-ci permettent l'applications de méthodes d'analyse statistique ainsi que multi-résolution adaptées à la structure et à la topologie de l'espace des rotations, ce qui est difficilement le cas pour les angles d'Euler.

Les taux de compression permis par la méthode proposée sont parfois étonnament élevés (50:1, 100:1, voire plus), tout en fournissant des animations reconstruites de qualité correcte, voire très bonne. Des taux de compression encore plus élevés peuvent sans doute être atteints par ajustement plus fin des paramètres de compression.

Il convient de remarquer que contrairement aux méthodes de compression d'animation squelettiques existantes (*cf.* partie 2.1), nous *ne quantifions pas* les données sur 16 bits afin d'accroître artificiellement le taux de compression d'un facteur 2 comme c'est le cas dans [Ari06] et [BPvdP07]. Néanmoins, il existe de nombreux points sur lesquels des améliorations sont nécessaires. Les principaux sont développés ci-dessous.

4.2.1 Compression

La méthode de compression présentée souffre de plusieurs défauts. Tout d'abord l'absence de métrique solide d'évaluation du résultat d'un point de vue qualitatif pose problème. Les meilleures métriques existantes sont souvent lourdes à mettre en place (nécessité d'apprentissage sur de larges bases) pour des résultats pas toujours satisfaisants. L'évaluation de la qualité du résultat doit donc souvent se faire par un être humain, ce qui limite d'autant les possibilités de comparaisons objectives des différentes méthodes de compression.

La donnée d'une telle métrique permettrait en outre de définir précisement le degré d'erreur commis en compressant, pour envisager l'adaptation automatique des paramètres de compression en fonction d'une borne sur l'erreur commise. Les paramètres de compression sont de surcroît relativement peu intuitifs à manipuler, c'est pourquoi un système automatique de choix des paramètre en fonction de critères plus naturels serait grandement nécéssaire.

D'autre part, du point de vue de la compression en elle même, nous avons remarqué (*cf.* partie 3.6) que l'AGP sur un niveau de détail de l'animation génère de l'ordre de deux fois plus de vecteurs propres que l'AGP sur une version mono-résolution, ceci pour une même variance conservée. Les causes de ce phénomène sont encore incertaines, et des travaux approfondis sur la représentation des niveaux de détails pourrait fournir des pistes pour corriger ce problème. De meilleurs résultats au niveau de l'AGP par niveaux impliquent évidemment des taux de compression encore meilleurs.

Il reste que la compression par multi-résolution constitue la *principale* source de compression, et permet d'obtenir des résultats bien meilleurs qu'en utilisant la compression par AGP seule. L'AGP sur les niveaux de détails ne constitue qu'un moyen d'*améliorer* les scores de la compression multi-résolution, en donnant une représentation plus compacte des niveaux de détail.

Enfin, la méthode proposée offre de meilleurs taux de compression sur des mouvements longs que sur des mouvements d'une courte durée(à fréquence d'échantillonnage égale), ces derniers engendrant moins de niveaux de détails.

4.2.2 Artéfacts visuels

Étant donné que nous travaillons sur les angles des articulations, les erreurs commises lors de la reconstruction se cumulent le long de la hiérarchie sur squelette, c'est pourquoi les extrêmités du squelettes présentent souvent des trajectoires erronées. Ce genre de défaut n'est en général pas gênant visuellement, sauf dans le cas où des contacts doivent être maintenus, comme dans cas des pieds sur le sol. Les pieds ont alors tendance à flotter ou glisser sur le sol, ce qui consitue un artefact visuel très remarquable.

Ce genre de défaut est bien connu, et de nombreuses techniques existent pour le corriger, par exemple par cinématique inverse. Pour ce faire, plusieurs stratégies existent :

a) Détecter automatiquement lorsque le pied touche le sol, et se servir de la position détectée afin d'effectuer la cinématique inverse pour les images suivantes [IAF06]

b) Encoder directement la correction de position à appliquer à chaque extrêmité, puis effectuer la cinématique inverse en utilisant la position corrigée [BPvdP07]

Nous pensons que d'un point de vue qualitatif, l'option b) semble la plus raisonnable, même si cela augmente le nombre de données à stocker. Les taux de compressions permis par la méthode authorisent l'ajout de quelques informations supplémentaires. Qui plus est, les deux signaux de correction ajoutés peuvent eux-même être compressés par une méthode adaptée, pour peu qu'ils soient reconstruits de manière exacte lors des contacts des pieds avec le sol.

D'autre part, il est probable que l'analyse des corrélations des orientation des joints du squelette fournisse des informations utiles à la cinématique inverse, notemment en réduisant l'espace de recherche lors du processus d'optimisation.

Nous envisageons également une adaptation des paramètres de compression en fonction de la profondeur d'un joint dans la hiérarchie, ceci afin de limiter les erreurs commises en amont pour ainsi améliorer le résultat final.

4.2.3 Préservation des discontinuités C^1

Dans le même ordre d'idée, les mouvements très brusques et/ou saccadés sont lissés par l'utilisation d'ondelettes splines. Pourtant ces types de détails (qui incluent les pas au sol) peuvent constituer des éléments cruciaux de l'animation, et leur conservation améliorerait grandement la qualité générale de la reconstruction.

Plusieurs pistes sont envisageables pour parvenir à cela. Nous avons tout d'abord pensé à l'applications de petites "rustines" (des *patches*) de corrections, constitués de petites pyramides de niveaux de détails applicables à la version multi-résolution afin de corriger localement le résultat obtenu. Ce type de solution est également envisageable pour la correction du *footskating*, même si celui-ci découle également du culmul des erreurs le long de la hiérarchie.

Une autre piste à envisager est l'emploi d'autres ondelettes pour l'analyse multi-résolution du mouvement. Nous songeons notemment à des ondelettes présentant le type de discontinuités recherchées, pouvant être utilisées localement pour reproduire le comportement du mouvement.

4.2.4 Analyse statistique

Tout d'abord l'approximation de l'analyse en géodésiques principales utilisée dans la présente méthode n'est qu'une linéarisation locale des données de rotation, même s'il s'agit de la "meilleure" possible. De récents travaux proposent cependant une formulation exacte du problème de l'AGP, dont certains résultats sont surprenants. On notera en particulier qu'une AGP exacte de données réelles n'engendre pas un nombre *a priori* borné de géodésiques décrivant le signal, contrairement à l'approximation de Fletcher qui en génère 3 dans tous les cas. De tels résultats suggèrent que les techniques d'analyse statistique linéaires ne sont peut-être pas adaptées au cas non-linéaire, et que des outils spécifiques leurs seraient préférables.

Les possibilités offertes par de tels outils sont séduisantes : une méthode puissante de tirage aléatoire d'orientations selon une distribution de probabilité permettrait une synthèse aléatoire des niveaux de détails les plus élevés de l'analyse multi-résolution, pour ainsi n'encoder que les paramètres de la distribution, tout en préservant les informations des hautes fréquences qui donnent son *style* à l'animation.

Le cadre probabiliste non-linéaire est cependant encore jeune et une meilleure compréhension de ce dernier semble nécéssaire à l'élaboration d'outils statistiques réellement adaptés, pratiques et puissants.

Bibliographie

[Ale91] R. M. Alexander. Optimum timing of muscle activation for simple models of throwing. *J. Theor. Biol.*, (150):349–372, 1991.

[Ari06] Okan Arikan. Compression of motion capture databases. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 890–897, New York, NY, USA, 2006. ACM Press.

[BF01] Samuel R. Buss and Jay P. Fillmore. Spherical averages and applications to spherical splines and interpolation. *ACM Trans. Graph.*, 20(2):95–126, 2001.

[BPvdP07] Philippe Beaudoin, Pierre Poulin, and Michiel van de Panne. Adapting wavelet compression to human motion capture clips. In *Graphics Interface 2007*, pages –, May 2007.

[DK00] J.J. Duistermaat and J. A. C. Kolk. *Lie Groups*. Springer, 2000.

[FLJ03] P. Thomas Fletcher, Conglin Lu, and Sarang C. Joshi. Statistics of shape via principal geodesic analysis on lie groups. In *CVPR (1)*, pages 95–101, 2003.

[FLPJ04] P. Thomas Fletcher, Conglin Lu, Stephen M. Pizer, and Sarang C. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Trans. Med. Imaging*, 23(8):995–1005, 2004.

[GMHP04] Keith Grochow, Steven L. Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 522–531, New York, NY, USA, 2004. ACM Press.

[Gra98] F. Sebastian Grassia. Practical parameterization of rotations using the exponential map. *journal of graphics tools*, 3(3):29–48, 1998.

[IAF06] Leslie Ikemoto, Okan Arikan, and David Forsyth. Knowing when to put your foot down. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 49–53, New York, NY, USA, 2006. ACM Press.

[Kar77] H. Karcher. Riemannian center of mass and mollifier smoothing. *Comm. Pure Appl. Math.*, 30:509–541, 1977.

[Ken90] W.S. Kendall. Probability, convexity, and harmonic maps with small image i: uniqueness and fine existence. *Proc. London Math. Soc.*, 61(2):371–406, 1990.

[Len99] Jerome Edward Lengyel. Compression of time-dependent geometry. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 89–95, New York, NY, USA, 1999. ACM Press.

[LS01] Jehee Lee and Sung Yong Shin. A coordinate-invariant approach to multiresolution motion analysis. *Graph. Models*, 63(2):87–105, 2001.

[Moa02] Maher Moakher. Means and averaging in the group of rotations. *SIAM J. Matrix Anal. Appl.*, 24(1):1–16, 2002.

[Pen99] X. Pennec. Probabilities and statistics on riemannian manifolds: Basic tools for geometric measurements, 1999.

[Pen06] Xavier Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *J. Math. Imaging Vis.*, 25(1):127–154, 2006.

[RDS+05] Inam Ur Rahman, Iddo Drori, Victoria C. Stodden, David L. Donoho, and Peter Schroder. Multiscale representations for manifold-valued data. *Multiscale Modeling Simulation* , 4(4):1201–1232, 2005.

[RP03] Paul S. A. Reitsma and Nancy S. Pollard. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 537–542, New York, NY, USA, 2003. ACM Press.

[RPE+05] Liu Ren, Alton Patrick, Alexei A. Efros, Jessica K. Hodgins, and James M. Rehg. A data-driven approach to quantifying natural human motion. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1090–1097, New York, NY, USA, 2005. ACM Press.

[Sho85] Ken Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, New York, NY, USA, 1985. ACM Press.

[SHP04] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 514–521, New York, NY, USA, 2004. ACM Press.

[SSK05] Mirko Sattler, Ralf Sarlette, and Reinhard Klein. Simple and efficient compression of animation sequences. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217, New York, NY, USA, 2005. ACM Press.

[Swe98] Wim Sweldens. The lifting scheme: a construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1998.

Euuh, non, rien.