



**HAL**  
open science

# Utilisation des relations spatiales pour l'analyse et l'édition de mouvement

Thibaut Le Naour

► **To cite this version:**

Thibaut Le Naour. Utilisation des relations spatiales pour l'analyse et l'édition de mouvement. Synthèse d'image et réalité virtuelle [cs.GR]. Université de Bretagne Sud, 2013. Français. NNT: . tel-00916514v1

**HAL Id: tel-00916514**

**<https://theses.hal.science/tel-00916514v1>**

Submitted on 10 Dec 2013 (v1), last revised 14 Mar 2014 (v2)

**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.



**THESE / UNIVERSITE DE BRETAGNE SUD**

**UFR Sciences et Sciences de l'Ingénieur**

*sous le sceau de l'Université Européenne de Bretagne*

Pour obtenir le grade de :

**DOCTEUR DE L'UNIVERSITE DE BRETAGNE SUD**

*Mention : STIC*

**Ecole Doctorale SICMA**

présentée par

**Thibaut LE NAOUR**

**IRISA-UBS Laboratoire de Recherche Informatique et ses  
Applications de Vannes et Lorient**

# **Utilisation des relations spatiales pour l'analyse et l'édition de mouvement**

**Thèse soutenue le 09 Décembre 2013,**  
devant la commission d'examen composée de :

**M. Frank Multon**

Professeur, IRISA/M2S Rennes, France / *President*

**Mme. Marie-Paule Cani**

Professeur, Inria Grenoble, France / *Rapporteur*

**M. Ronan Boulic**

Maître d'Enseignement et de Recherche, EPFL Lausanne, Suisse / *Rapporteur*

**M. Frank Hetroy**

MCF, Inria Grenoble, France / *Examinateur*

**Mme. Sylvie Gibet**

Professeur, IRISA-UBS, France / *Directeur*

**M. Nicolas Courty**

HDR, IRISA-UBS, France / *Encadrant*



# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>1 Introduction</b>	<b>3</b>
1.1 L'animation de personnage en 3D	3
1.2 Limites du processus classique	7
1.3 Contributions	9
1.4 Organisation du manuscrit	10
<b>I Le mouvement dans l'espace métrique</b>	<b>13</b>
<b>2 Géométrie des distances appliquée au mouvement</b>	<b>15</b>
2.1 Introduction	15
2.2 Représentation du mouvement	17
2.3 Le problème de la géométrie des distances (DGP)	18
2.3.1 Méthodes de résolution classiques	19
2.3.2 Applications	25
2.4 Conclusion et application au mouvement	27
<b>3 Cinématique guidée par les distances</b>	<b>31</b>
3.1 Introduction	31
3.2 Approches connexes	32
3.3 Cinématique guidée par les distances : formalisation	34
3.3.1 DGP appliqué à l'animation de personnage	34
3.3.2 Formulation par un problème de minimisation	35
3.4 Méthodes d'optimisation pour résoudre un système d'équations	35
3.5 Adaptation de mouvement et cinématique inverse	36
3.6 Résultats	38
3.6.1 Suivi de mouvement	38
3.6.2 Cinématique inverse	39
3.6.3 Modification de mouvement (cible à atteindre pendant la marche)	42
3.6.4 Application à un graphe cycle	45
3.7 Discussion et perspectives	46
<b>4 Application à la recherche de mouvement</b>	<b>47</b>
4.1 Introduction	47
4.2 Travaux connexes	48
4.3 Représentation du mouvement	49
4.4 Méthodologie	50

4.5	Résultats . . . . .	50
4.6	Conclusion . . . . .	52
<b>II Représentation Laplacienne du mouvement associée aux distances comme contrainte implicite</b>		<b>55</b>
<b>5</b>	<b>L'opérateur Laplacien pour l'animation 3D</b>	<b>57</b>
5.1	Introduction . . . . .	57
5.2	Présentation de l'opérateur Laplacien . . . . .	58
5.2.1	L'opérateur Laplacien continu . . . . .	58
5.2.2	L'opérateur Laplacien discret : application à l'édition de graphe . . . . .	59
5.3	Applications à l'informatique graphique . . . . .	66
5.3.1	Traitement de maillage . . . . .	66
5.3.2	Édition de mouvement . . . . .	72
5.4	Conclusion . . . . .	75
<b>6</b>	<b>Couplage des distances avec l'opérateur Laplacien</b>	<b>77</b>
6.1	Introduction . . . . .	77
6.2	Préservation des distances . . . . .	78
6.2.1	Présentation du problème . . . . .	78
6.2.2	Résolution par une technique de minimisation alternée . . . . .	80
6.3	Application à l'édition de graphe . . . . .	83
6.3.1	Résolution numérique . . . . .	83
6.3.2	Exemple simple . . . . .	83
6.4	Conclusion . . . . .	83
<b>7</b>	<b>L'opérateur Laplacien 3D+t pour une édition spatio-temporelle du mouvement</b>	<b>85</b>
7.1	Introduction . . . . .	85
7.2	Approches connexes . . . . .	88
7.3	Utilisation de l'opérateur 3D+t pour l'édition de mouvement . . . . .	88
7.3.1	L'opérateur discret Laplacien 3D+t . . . . .	88
7.3.2	Du mouvement capturé vers le graphe 3D+t . . . . .	89
7.3.3	Couplage des distances avec l'opérateur Laplacien pour l'édition du mouvement . . . . .	89
7.3.4	Poids Laplacien . . . . .	91
7.3.5	Contraintes de rotation . . . . .	92
7.4	Application à l'édition de mouvement . . . . .	93
7.4.1	Édition de mouvement . . . . .	93
7.4.2	Adaptation de mouvement . . . . .	96
7.5	Disussion et perspectives . . . . .	97

---

<b>8</b>	<b>Application à la reconstruction de trajectoires de marqueurs</b>	<b>99</b>
8.1	Introduction . . . . .	99
8.2	Approches connexes . . . . .	100
8.2.1	Méthodes dédiées à la reconstruction de mouvement . . . . .	100
8.2.2	Utilisation de l'opérateur Laplacien pour de l'interpolation de données . . . . .	102
8.3	Constats sur le mouvement capturé . . . . .	102
8.4	Reconstruction du mouvement par la préservation des relations spatiales . . . . .	105
8.5	Résultats . . . . .	106
8.6	Conclusion et perspectives . . . . .	109
<b>9</b>	<b>Contrôle de maillage temps réel guidé par des marqueurs issus de la capture de mouvement</b>	<b>111</b>
9.1	Introduction . . . . .	111
9.2	Approches connexes . . . . .	114
9.2.1	Déformation de maillage . . . . .	114
9.2.2	Animation de maillage . . . . .	115
9.2.3	Notre approche . . . . .	120
9.3	Présentation du système . . . . .	120
9.4	Déformation du maillage volumique incluant les propriétés du squelette . . . . .	122
9.4.1	Couplage squelette/maillage via la construction du maillage volumique . . . . .	122
9.4.2	Déformation de maillage As-Rigid-As-Possible . . . . .	123
9.4.3	Préservation des propriétés du squelette . . . . .	127
9.4.4	Projection du squelette dans l'espace des postures admissibles . . . . .	129
9.4.5	Relâchement des contraintes d'ARAP autour des articulations . . . . .	133
9.4.6	Déformation finale du maillage de référence . . . . .	135
9.4.7	Algorithme . . . . .	136
9.5	Implémentation et résultats . . . . .	136
9.5.1	Tests basiques . . . . .	137
9.5.2	Édition de maillage guidée par des sommets définis manuellement . . . . .	138
9.5.3	Animation de maillage guidée par le squelette . . . . .	140
9.5.4	Animation de maillage guidée par les marqueurs . . . . .	142
9.6	Conclusion et perspectives . . . . .	144
<b>10</b>	<b>Conclusion</b>	<b>147</b>
10.1	Contributions et discussions . . . . .	147
10.2	Perspectives . . . . .	149

---

<b>Bibliographie</b>	<b>151</b>
<b>Annexes</b>	<b>166</b>
<b>A Méthode de résolution de Newton-Raphson</b>	<b>169</b>
<b>B Technique <i>As-Rigid-As-Possible</i></b>	<b>171</b>







# 1

## Introduction

### Sommaire

---

<b>1.1 L'animation de personnage en 3D</b> . . . . .	<b>3</b>
<b>1.2 Limites du processus classique</b> . . . . .	<b>7</b>
<b>1.3 Contributions</b> . . . . .	<b>9</b>
<b>1.4 Organisation du manuscrit</b> . . . . .	<b>10</b>

---

### 1.1 L'animation de personnage en 3D

Ce manuscrit traite de l'animation de personnage guidée par des données capturées. Cette thématique, dont l'objectif principal est de restituer virtuellement le mouvement d'un acteur capturé dans un contexte réel, fait l'objet de nombreuses études et applications. Parmi celles-ci, nous pouvons distinguer :

- **L'industrie du film** : dans ce contexte, l'objectif est d'animer avec réalisme des humanoïdes ou des créatures reproduisant une gestuelle humaine, c'est par exemple le cas des films *Avatar* ou *Le pôle express* (figure 1.1 (a) et (b)).
- **L'industrie du jeu vidéo** : la capture de mouvement humain est utilisée pour obtenir des gestes techniques précis et réalistes difficiles à reproduire par des animateurs, cela concerne par exemple les jeux de sports tel que la série *FIFA* (figure 1.1 (c)). Elle permet également d'acquérir à bon marché un grand nombre d'animations.
- **Domaines d'application connexes** : en dehors de la reproduction du mouvement d'un personnage, l'étude du mouvement humain capturé est également exploitée dans le cadre d'autres types d'applications telles que l'analyse de performance sportives (e.g [90, 186]), la classification, ou la reconnaissance de gestes (par exemple dans le cadre de la reconnaissance de gestes de langue des signes : e.g [58, 91], la danse : e.g [112, 172] ou les gestes musicaux [23, 133])

Pour obtenir numériquement le mouvement capturé, plusieurs techniques d'acquisition sont actuellement utilisées : la *mocap optique*, technique que nous utilisons dans cette étude, est la plus répandue dans l'industrie pour sa fiabilité et sa précision. Son principe est de mesurer



FIGURE 1.1 – Animation de personnage guidée par des données capturées dans les films d’animation et les jeux.

l’évolution dans l’espace et dans le temps avec des caméras infrarouges le déplacement de marqueurs réfléchissants. D’abord dédiée à l’enregistrement des mouvements du corps, elle permet aujourd’hui de capturer l’animation du visage. La *mocap gyroscopique* se base sur l’utilisation de capteurs gyroscopiques et inertiels intégrés à des combinaisons portées par les comédiens. Elle présente l’avantage de ne pas nécessiter de caméra, ni d’émetteur ou de marqueur, cela implique que le système ne produit pas d’occultations mais ne prend pas en considération les informations liées à l’environnement. La *mocap sans marqueurs* qui est associée à des techniques plus récentes, repose sur des méthodes d’acquisition utilisant plusieurs caméras simultanément. Ces techniques présentent l’avantage de produire des captures avec des systèmes moins intrusifs que les autres mais sont toutefois moins précises que celles utilisant des marqueurs.

De manière complémentaire à la capture du mouvement brut, une chaîne de traitement des données permettant d’obtenir l’animation virtuelle d’un personnage est nécessaire. En voici une description synthétique.

**Processus d’animation de personnage** Classiquement, le processus d’animation d’un personnage en 3D peut être caractérisé par trois phases, chacune étant associée à un niveau différent de représentation des données :

**1. Acquisition des trajectoires des marqueurs** Comme présenté dans la figure 1.2, la capture de mouvement s’accompagne de plusieurs traitements. Dans la plupart des systèmes, comme par exemple *Vicon* [175] ou *Qualisys* [1], des marqueurs sont positionnés sur l’acteur à des endroits spécifiques du corps, généralement près des articulations. C’est à partir de caméras émettant un rayonnement infrarouge que sont calculées les positions des marqueurs en trois dimensions dans l’espace Cartésien. Plus précisément, les données fournies par les caméras permettent d’estimer par triangulation la position dans l’espace de la scène. Le nombre

de caméras dépend de la nature des mouvements, de leur précision et de leur rapidité. Enfin l'étape de calibration (étalonnage du système avec un modèle dont les caractéristiques sont connues) est suivie de l'acquisition des trajectoires des marqueurs.

A l'issue de l'étape de capture de mouvement, nous obtenons les positions des marqueurs dans l'espace et le temps. Il est nécessaire également d'étiqueter les différents marqueurs, et d'effectuer des opérations de correction et de post-traitement dues à :

- l'inversion de marqueurs pendant certaines phases d'enregistrement ;
- l'existence de trajectoires manquantes dues à des occultations. Cela arrive par exemple pour l'animation de la main qui implique le déplacement d'un grand nombre de marqueurs dans un espace réduit ;
- l'imprécision et le bruit des données nécessitant une étape de filtrage.

Enfin, la fréquence d'échantillonnage est choisie par l'utilisateur et dépend grandement des mouvements capturés. On prendra par exemple 30 *Hz* par seconde pour une capture de basse résolution (mouvements lents) et jusqu'à 500 *Hz* par seconde pour de la haute résolution (gestes rapides, comme les gestes musicaux par exemple). En général, ces informations sont stockées sous la forme d'un ensemble de positions dans le temps dans des formats dédiés (*C3D*, *TSV*, ...).

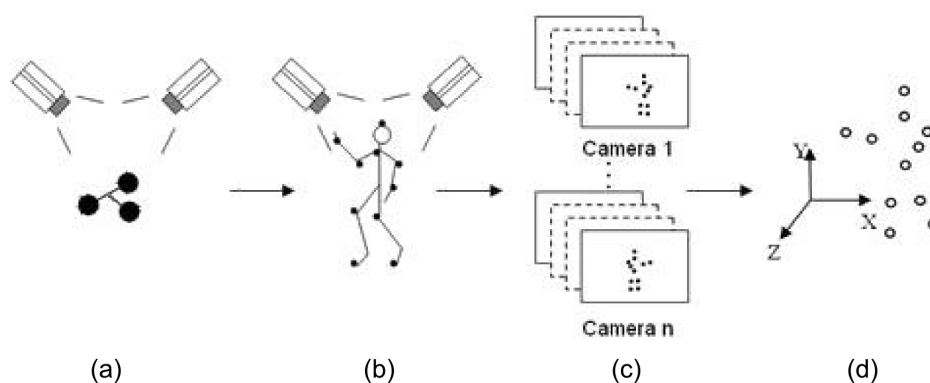


FIGURE 1.2 – Processus de capture du mouvement : il comprend une phase de calibration (a), la capture de l'acteur (b), puis une mise en commun des trajectoires capturées associées aux caméras (c). Enfin en (d), le système calcule les positions des marqueurs en 3 dimensions (avec éventuellement divers traitements comme la reconstruction de trajectoires manquantes, etc.).

**2. Reconstruction de l'animation du squelette** Une fois les trajectoires des marqueurs obtenues, il est classique de chercher à obtenir une représentation du mouvement sous la forme de l'animation d'un squelette. Cette nouvelle représentation se décrit comme une structure arborescente composée d'un ensemble d'articulations, chacune possédant une orientation et une position. La mise à jour de ces orientations au cours du temps produit l'animation du squelette. Afin d'obtenir cette nouvelle représentation du mouvement, plusieurs étapes sont requises :

- Calcul des centres articulaires : cette étape passe par l'identification de groupes de marqueurs dont la transformation d'une posture à une autre est commune. Ainsi il est possible d'extraire les positions et les rotations associées aux articulations.
- Reconstruction des données manquantes : la première phase fournit une estimation de plusieurs centres articulaires mais en général il s'agit des articulations les plus importantes. En effet positionner plusieurs marqueurs pour chaque articulation, comme par exemple pour les mains, rend le travail de post-traitement décrit dans le paragraphe précédent très fastidieux. Afin de retrouver les rotations associées à ces articulations, l'animateur a le plus souvent recours à des techniques de cinématique inverse. Il s'agit d'estimer par une méthode d'optimisation la trajectoire incomplète de l'articulation considérée. Ces deux dernières étapes sont illustrées dans la figure 1.3 (de (a) vers (b)).

L'ensemble de ces traitements est souvent opéré via des logiciels tels que *MotionBuilder*, *3DSMax* ou encore *Maya*. Enfin, les formats de stockage les plus connus sont *BVH*, *BIP* et *ASF/AMC*.

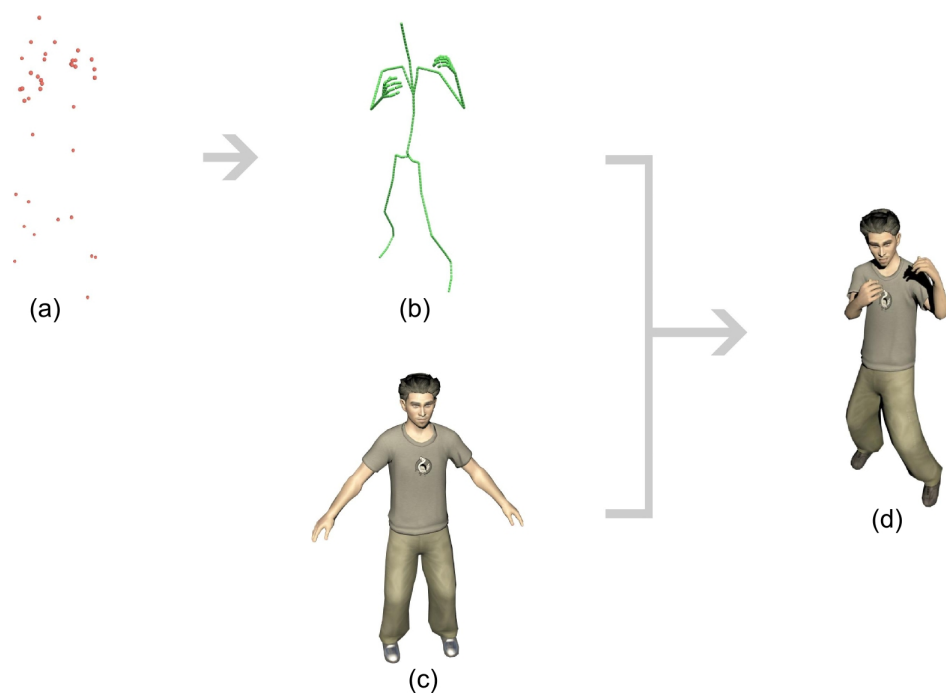


FIGURE 1.3 – Processus d'animation de maillage. Pour une même posture : en (a) le nuage de marqueurs issu de la capture de mouvement ; en (b) le squelette reconstruit dans l'espace des rotations ; la figure (c) représente le maillage correspondant à la posture de référence (*bind pose*) ; en (d) le maillage animé guidé par l'animation du squelette.

**3. Animation du maillage** L'objectif final d'une application 3D est bien sûr d'animer un maillage. Dans ce contexte, le mouvement est obtenu par la déformation du maillage guidée par l'animation du squelette, chaque maillage étant défini par une information géométrique et

topologique. De manière classique, la déformation du maillage est directement guidée par le squelette. Plusieurs manipulations sont requises :

- Adaptation du squelette : le maillage à animer correspond dans la majorité des cas à une morphologie différente du squelette précédemment reconstruit. Il est donc impératif de passer par une opération qui consiste à adapter le mouvement du squelette vers la nouvelle morphologie. A noter qu'il est également nécessaire d'avoir une posture commune entre la géométrie et le squelette, posture classiquement appelée la *bind pose*.
- Étape de *rigging* : pour animer le maillage, il est nécessaire de "l'attacher" à un squelette ; cela revient à établir un lien paramétré par un poids entre chacun des sommets du maillage et une ou plusieurs articulations du squelette. Cette manipulation appelée *rigging*, est en général opérée manuellement. La véritable difficulté réside dans la détermination des poids à associer aux articulations. Cette étape est particulièrement importante relativement à la qualité des résultats obtenus (elle correspond à la mise en relation du squelette (b) avec le maillage en posture de référence (c) dans la figure 1.3).
- *Skining* : une fois le *rigging* effectué, il existe un grand nombre de techniques pour obtenir des déformations plus ou moins réalistes. La technique la plus connue, appelée "linear blend skinning" (Lewis [105]), est rapide mais peu réaliste. D'autres méthodes, comme celle s'appuyant sur les quaternions duaux [93] permettent des résultats visuellement plus réalistes.
- Effet final : enfin, plusieurs études se consacrent à l'amélioration du rendu final ; il s'agit de modifier le *skining* classique en cherchant à mieux préserver le volume [142] ou en intégrant de nouveaux éléments comme par exemple les muscles via des modèles géométriques ou physiques. On peut aussi ajouter l'animation des vêtements, des cheveux, etc.

## 1.2 Limites du processus classique

Dans le processus précédemment décrit, nous avons mis en évidence la démarche classique consistant à se ramener à une représentation par squelette à partir des trajectoires des marqueurs pour animer le maillage. Le mouvement par squelette correspond à l'hypothèse de rigide poly-articulé paramétré par des transformations rigides de l'espace (des rotations la plupart du temps). Utiliser cette représentation offre plusieurs avantages : elle préserve implicitement la longueur des segments, admet une paramétrisation naturelle du mouvement et permet ainsi de caractériser facilement les butées articulaires, mais aussi de décrire le mouvement avec moins de paramètres. Ainsi il existe de nombreuses techniques pour synthétiser et éditer un mouvement en travaillant sur les rotations comme par exemple la cinématique inverse, ou encore les techniques utilisant des systèmes physiques. Toutefois, en appliquant le processus classique précédemment décrit dans le cadre d'une application traitant de la langue des signes, application requérant une grande précision et rapidité des mouvements, nous avons constaté que cette représentation présentait un certain nombre d'inconvénients.

A travers le projet SIGNCOM [11, 58], dont l'objectif était d'établir le dialogue en langue des signes française (LSF) entre deux personnages réel et virtuel, nous avons reproduit le processus classique d'animation de personnage à partir de données capturées. Dans ce contexte, il était nécessaire de produire une animation de qualité pour respecter le sens des signes produits. Or nous avons identifié un certain nombre de problèmes altérant la qualité et potentiellement la sémantique des animations produites [58]. Les problèmes soulevés, illustrés par la figure 1.4, concernent surtout l'animation des mains ainsi que le respect du contact entre certaines parties du corps. Le premier cas 1.4(a) met en évidence l'absence de contact entre les deux doigts de la main : cet exemple illustre particulièrement bien le fait que le résultat obtenu est très approximatif et n'est pas satisfaisant d'un point de vue sémantique. Le deuxième exemple 1.4(b) laisse apparaître un grand nombre d'intersections inter-maillage ainsi qu'une mauvaise estimation du *rigging* (en particulier au niveau du pouce).

Les raisons techniques qui ont amené à cet ensemble de constats peuvent être attribuées à différentes causes :

- **Calcul des centres articulaires** : cette étape est efficace pour mesurer le mouvement des membres du corps mais difficile à mettre en oeuvre pour les articulations nécessitant de positionner plusieurs marqueurs proches les uns des autres. Par exemple, capturer le mouvement des mains qui comportent chacune au moins vingt deux degrés de liberté nécessite de poser suffisamment de marqueurs au niveau des articulations. Cela peut engendrer des occultations de certains marqueurs, et par conséquent des post-traitements fastidieux. Toutes ces raisons font que la reconstruction du mouvement des mains peut s'avérer approximative (dans la figure 1.4(b), il apparaît qu'une telle approximation implique occasionnellement des collisions entre les doigts).
- **Représentation de dimension réduite** : en réduisant l'espace de représentation au squelette, le processus de calcul des centres articulaires perd l'information de distance entre le centre articulaire et les marqueurs positionnés à la surface de la peau. Ainsi le squelette, bien que donnant une bonne estimation des proportions osseuses, ne prend pas en considération le type de morphologie du modèle : est-il musclé, mince, ou corpulent ? Cette information peut être importante, comme par exemple lors du contact entre deux doigts.
- **Représentation par squelette** : la représentation par squelette en forme d'arbre met uniquement en relation les articulations qui ont des liens de parenté entre elles. Lors de la correction ou de l'édition, cette représentation pose le problème de la préservation des autres relations existantes entre les différents éléments d'une même scène (le reste du corps, d'autres personnages ou l'environnement). Par exemple, il n'est pas rare après une opération de redimensionnement du squelette (*retargetting*) de voir apparaître des intersections entre les segments du squelette lors du mouvement. Inversement, dans la figure 1.4, nous pouvons observer, après la phase de *retargetting*, la perte du contact entre deux doigts. Ceci est lié au fait que la représentation par squelette ne prend pas en considération les relations de voisinage et de proximité entre les articulations.



FIGURE 1.4 – Exemples d’incohérences issues du processus d’animation de maillage.

- **Étape de *Rigging*** : cette étape, souvent opérée par un graphiste, est fastidieuse et dépend grandement de ses compétences. Elle est particulièrement difficile dans les zones comportant beaucoup d’articulations, où un même sommet peut être guidé par de nombreuses articulations. C’est le cas de la main illustrée par la figure 1.4(b), où le *rigging* des pouces est approximatif.

**Problématiques** Ces différentes limites nous amènent à nous poser les deux questions suivantes :

- **existe-t’il d’autres espaces de représentation que l’espace des rotations permettant l’édition, le redimensionnement ou la correction d’un mouvement issu de données capturées ?** De telles représentations devraient considérer les relations spatiales entre les articulations, les interactions entre personnages ou avec l’environnement ;
- d’une manière plus générale, **le processus classique pour animer des personnages à partir de données capturées peut-il être modifié ou simplifié pour limiter la complexité des post-traitements et éviter les erreurs de reconstruction ?** Un tel processus devrait en particulier permettre une amélioration significative de la qualité de l’animation par rapport à la chaîne de production classique.

### 1.3 Contributions

Dans cette thèse nous nous intéressons à l’analyse et l’édition de mouvement de personnages, ceci dans le but de corriger, modifier ou adapter le mouvement capturé à de nouveaux contextes. Par exemple on peut modifier spatialement certaines trajectoires, en intégrant interactivement le mouvement dans un nouvel environnement ou encore en redimensionnant le squelette. Dans ce contexte, nous proposons d’étudier d’autres représentations du mouvement



par le biais d'un ensemble de relations spatiales. Nos contributions se décomposent autour de deux axes d'étude :

Contrairement aux méthodes classiques qui utilisent une représentation arborescente pour caractériser une posture du mouvement, nous proposons de représenter les squelettes par un ensemble de distances. Cette représentation permet d'exprimer une articulation par sa relation spatiale avec les autres articulations du squelette mais également avec l'environnement. Cette propriété permet d'une part de préserver les relations spatiales entre les articulations durant l'édition et d'autre part de contrôler plus simplement et plus intuitivement le mouvement. Au delà de la problématique de cette thèse, nous évaluons la capacité d'une telle représentation à être exploitée pour l'analyse de mouvement ; cela donne lieu à une application de recherche de mouvements dans des grandes bases de données.

Nous proposons dans un second temps de définir le mouvement par un ensemble de graphes dont les sommets sont caractérisés par une information différentielle. A travers cette représentation, nous proposons une nouvelle méthode d'édition du mouvement couplant des contraintes de distance avec l'opérateur Laplacien discret. Plus précisément, un sommet est exprimé par sa relation spatiale avec ses sommets voisins et éventuellement par une distance fixe à un autre sommet du graphe. A travers cette méthode, nous proposons trois applications dédiées à la reconstruction et l'édition de mouvement : (i) l'édition interactive d'animation de squelette, où nous proposons d'éditer tout type de mouvement avec de fortes déformations tout en préservant l'information spatio-temporelle ; (ii) la reconstruction de trajectoires de marqueurs : en faisant l'hypothèse qu'il existe un lien entre la trajectoire d'un marqueur et celles de ses voisins, nous reconstruisons les trajectoires incomplètes ; (iii) l'animation de maillage où nous proposons un nouveau processus d'animation directement guidé par les trajectoires des marqueurs.

## 1.4 Organisation du manuscrit

Ce manuscrit s'organise en deux parties.

La première partie **I** traite de l'analyse et de l'édition du mouvement dans l'espace métrique. Dans le premier chapitre **2**, nous introduisons le problème général de l'animation de squelette dans l'espace métrique. Nous verrons que cette nouvelle formulation du squelette est similaire au problème classique de géométrie des distances (*DGP : Distance Geometry Problem*) et présenterons les différentes méthodes de résolution classiquement utilisées. Le chapitre suivant **3**, traitera de notre méthode d'édition de mouvement guidée par les distances. Nous verrons que cette nouvelle représentation permet de rejouer un mouvement en cinématique directe mais est également capable d'éditer ou de générer des animations. Nous verrons enfin dans le chapitre **4** que cette caractérisation du squelette dans l'espace métrique possède plusieurs propriétés intéressantes pour l'analyse de mouvement et en particulier dans le cadre de la recherche de mouvement dans des bases de données.

Dans la partie **II**, nous nous intéresserons à une représentation du mouvement dans l'espace Laplacien, à travers sa formulation par un ensemble de graphes. Dans un premier chapitre **5**

nous rappellerons les propriétés de l'opérateur Laplacien discret ainsi que les divers travaux connexes appliqués au domaine de l'informatique graphique. Dans le second chapitre 6 nous présenterons notre méthode d'édition de graphe. Nous remarquerons en particulier l'intérêt de coupler lors de l'édition du graphe la préservation de l'information différentielle avec le respect de contraintes de distance appliquées à certaines arêtes. Les chapitres suivants permettront d'appliquer sur des cas concrets notre méthode. Dans le chapitre 7, nous présenterons une nouvelle technique d'édition de mouvement permettant de préserver ses propriétés spatio-temporelles. Le chapitre suivant 8 sera appliqué à la reconstruction de trajectoires de marqueurs. Enfin nous présenterons dans le chapitre 9 une nouvelle méthode d'animation de maillage directement guidée par l'animation des marqueurs.

Enfin, nous conclurons sur les différentes approches et proposerons des perspectives d'améliorations et de travaux dans la continuité de cette thèse.



## **Première partie**

# **Le mouvement dans l'espace métrique**



# 2

## Géométrie des distances appliquée au mouvement

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>15</b>
<b>2.2</b>	<b>Représentation du mouvement</b>	<b>17</b>
<b>2.3</b>	<b>Le problème de la géométrie des distances (DGP)</b>	<b>18</b>
2.3.1	Méthodes de résolution classiques	19
2.3.2	Applications	25
<b>2.4</b>	<b>Conclusion et application au mouvement</b>	<b>27</b>

---

### 2.1 Introduction

Nous nous intéressons dans cette thèse à l'étude des relations spatiales entre les éléments caractérisant un personnage en mouvement. Dans le cas d'un squelette, il s'agit de l'étude des relations spatiales entre les articulations. Notre première approche consiste à caractériser ces relations spatiales par des distances entre les articulations. Plus précisément, à partir des distances extraites d'un mouvement capturé, notre objectif est à la fois de le **rejouer** dans l'espace métrique mais également de le contraindre à de nouvelles tâches pour de la **génération** ou de **l'édition** (modification d'un mouvement existant).

Traditionnellement, l'animation d'un personnage est représentée par une suite de squelettes dans le temps. Un squelette est une hiérarchie d'articulations entre lesquelles les distances sont fixes. Dans un problème de cinématique inverse, on cherche à déplacer un effecteur (typiquement l'extrémité de la chaîne) vers un point  $x$ . La chaîne du squelette d'animation peut-être représentée par un vecteur  $q$  décrivant les transformations (positions cartésiennes, angulaires) des différentes parties intervenant dans le mouvement. Dans un problème de cinématique directe, pour une chaîne à  $n$  articulations, on définit la fonction  $f : SO(3)^n \rightarrow \mathbb{R}^3$  telle que :

$$x = f(q). \tag{2.1}$$

Inversement dans un problème de cinématique inverse, on a :

$$\mathbf{q} = f^{-1}(\mathbf{x}), \quad (2.2)$$

où  $f$  représente l'ensemble des transformations qu'il faut appliquer à la chaîne  $\mathbf{q}$  pour déplacer l'effecteur en  $\mathbf{x}$ .

Dans cette thèse nous proposons une approche originale qui vise à représenter chaque posture clé d'un mouvement par un ensemble de distances et non par des informations de positions ou de coordonnées angulaires. Cette re-formulation du problème n'a jusqu'à présent jamais été utilisée, et nous verrons dans la suite de ce document qu'elle présente un certain nombre d'intérêts, à la fois d'un point de vue théorique et applicatif. Il s'agit alors de contrôler le mouvement d'un système articulé en spécifiant des relations de distance à vérifier entre différents points du système. En caractérisant le mouvement dans l'espace métrique, la chaîne du squelette est ainsi représentée par un vecteur  $\mathbf{w}$  décrivant un ensemble de positions associées à ces distances. Pour  $n$  articulations associées à  $m$  distances, nous pouvons exprimer la cinématique directe avec une nouvelle fonction  $h : \mathbb{R}^m \rightarrow \mathbb{R}^{3n}$  :

$$\mathbf{x} = h(\mathbf{w}). \quad (2.3)$$

Inversement pour la cinématique inverse, nous utilisons la notation :

$$\mathbf{w} = h^{-1}(\mathbf{x}). \quad (2.4)$$

Nous remarquons que cette formulation du problème peut être vue comme un problème de géométrie des distances (DGP pour *Distance Geometry Problem*) qui consiste à trouver les coordonnées d'un ensemble de points à partir d'un ensemble de distances entre ces points. Un tel problème fait aujourd'hui l'objet de nombreuses études comme par exemple en géolocalisation pour de la reconstruction de réseaux de capteurs ou en chimie pour la reconstruction de structures moléculaires.

**Intérêt de la méthode.** L'intérêt de la formulation proposée réside dans **la manière de définir les tâches à travers des distances**. Nous verrons dans le chapitre suivant 3, à travers un jeu de tests, que cette nouvelle façon d'exprimer les tâches simplifie grandement le travail de contrôle de l'animation, notamment en fournissant **une spécification implicite des contacts** (par opposition à une définition explicite spatiale des zones de contact). Ce dernier point fait l'objet de travaux récents [75, 76] qui montrent l'intérêt de la communauté pour cette problématique. Par ailleurs, notre système permet de traiter dans son ensemble le squelette, et la résolution d'une tâche ou de plusieurs tâches simultanées se fait simplement. Nous verrons enfin qu'à travers un découplage de l'information fixe (distances des segments du squelette) et de l'information qui est modifiée dans le temps (les autres distances), nous pouvons émettre des priorités entre les contraintes et ainsi donner un style à l'animation.

**Plan du chapitre.** Dans un premier temps nous présentons quelle représentation du mouvement nous avons choisie pour travailler dans l'espace métrique. Nous verrons par la suite

pourquoi cette représentation se formule comme un problème de DGP et quelles sont les méthodes existantes pour une résolution du problème. Enfin nous discuterons de la méthode à retenir pour l'appliquer à notre contexte avant de conclure.

## 2.2 Représentation du mouvement

Formulé dans l'espace métrique, le mouvement est exprimé comme l'évolution des distances euclidiennes entre les articulations du squelette au cours du temps. Tout comme la représentation classique du mouvement, nous considérons le mouvement dans ce nouvel espace comme une suite de postures indépendantes entre elles.

**Représentation d'une posture** Pour chaque posture, nous extrayons un graphe  $G = (\mathbf{V}, \mathbf{E}, d)$  où  $\mathbf{V}$  est l'ensemble de  $n$  sommets et  $\mathbf{E}$  est l'ensemble des arêtes évoluant dans le temps. Plus précisément  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$  est un vecteur de coordonnées dans  $\mathbb{R}^n$  qui contient les vecteurs  $\mathbf{v}_i$  tel que  $\mathbf{v}_i = (v_{i,1}, v_{i,2}, v_{i,3})^T$  est dans  $\mathbb{R}^3$ . Pour tout couple de sommets  $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbf{E}$ ,  $d$  est l'ensemble des distances Euclidiennes de  $\mathbf{V} \times \mathbf{V} \rightarrow \mathbb{R}_+$ . La structure spatiale du graphe est alors exprimée par le système d'équations :

$$\forall (\mathbf{v}_i, \mathbf{v}_j) \in \mathbf{E}, \|\mathbf{v}_i - \mathbf{v}_j\| = d_{ij}, \quad (2.5)$$

où  $\|\cdot\|$  est la norme euclidienne. L'extraction des distances à partir de données capturées se fait simplement pour chaque posture indépendamment les unes des autres.

Notons que cette représentation accroît considérablement la quantité de données caractérisant un squelette par rapport à une représentation par des rotations. Par exemple pour un squelette de 31 articulations, 465 valeurs de distances sont nécessaires contre 31\*3 valeurs dans l'espace des rotations. Remarquons par ailleurs que cette nouvelle représentation crée de la redondance dans l'information. Nous verrons en particulier dans le chapitre 4 que cette propriété est intéressante pour caractériser le mouvement.

**Formulation d'un problème de géométrie des distances** A partir de cette représentation d'un squelette, notre problème consiste à retrouver les positions des sommets à partir de l'ensemble de distances  $d$ . Cette formulation du problème est équivalente au problème connu de DGP (Problème de Géométrie des Distances) : à partir de l'ensemble des distances caractérisant le graphe  $G$ , trouver les positions des sommets. Dans la section suivante nous expliquons plus en détail ce qu'est le problème de DGP et dans quels contextes il est appliqué. Nous recensons également les différentes méthodes de résolution applicables à notre problème.



## 2.3 Le problème de la géométrie des distances (DGP)

Le concept de géométrie des distances a été introduit en 1928 par Menger [120] qui propose de caractériser plusieurs concepts de géométrie (e.g congruence, ensemble convexe) en terme de distances. Ces premières études complétées par les travaux de Blumenthal [20] ont donné lieu à la discipline connue sous le nom de *Distance Geometry* (DG).

Le problème de DGP (*Geometry Distance Problem*) s'applique généralement à plusieurs types de structures : protéines, réseaux, etc. et se décline en plusieurs sous problèmes : (i) les paires de distances entre éléments sont connues dans leur **intégralité** ou **partiellement**, (ii) les distances sont connues de manière **exacte** ou **approximative**. Dans le cas général, nous représentons ici ces structures comme des graphes et ne disposons a priori d'aucune estimation spatiale de la structure.

Nous considérons le graphe  $G$  décrit dans la section précédente. Soit  $\|\cdot\|$  la norme Euclidienne. Si les coordonnées  $\mathbf{v}_1, \dots, \mathbf{v}_n$  sont connues, les distances  $d_{i,j}$  entre les sommets  $i$  et  $j$  peuvent être calculées par  $d_{i,j} = \|\mathbf{v}_i - \mathbf{v}_j\|$ . Inversement, si les distances  $d_{i,j}$  sont données, les coordonnées des sommets  $\mathbf{v}_1, \dots, \mathbf{v}_n$  peuvent être également calculées, cependant la résolution du problème n'est pas directe. Le problème se traduit par : trouver les coordonnées des sommets  $\mathbf{v}_i$  telles que :

$$\|\mathbf{v}_i - \mathbf{v}_j\| = d_{i,j} \text{ pour } (\mathbf{v}_i, \mathbf{v}_j) \in S, \quad (2.6)$$

où  $S$  est l'ensemble des paires de sommets dont les distances sont connues. Ce dernier problème est couramment appelé *distance geometry problem* en mathématiques [20]. D'autres noms lui ont été donné comme *graph embedding problem* en science de l'informatique [145], ou encore *multidimensional problem* en statistiques [171] et *graph realisation problem* en théorie des graphes [70]. Bien que dans ce manuscrit nous nous intéressons à une résolution dans  $\mathbb{R}^3$ , le problème est applicable dans  $n$  dimensions. Ce type de problème a de nombreuses applications comme en reconstruction de protéine en chimie [153], en localisation de réseau de capteurs (*sensor network localization*) [17], en reconnaissance d'image [98] ou en classification de protéines [79].

Appliqué par exemple à la détermination de structures moléculaires, le problème de DGP est appelé MDGP. Dans ce contexte les sommets du graphe sont des atomes. En pratique, les distances sont issues d'expérimentations physiques ou d'estimations théoriques (comme par exemple issues de données de résonance magnétique numérique (NMR) [127]), les valeurs sont donc approximatives. Le problème peut être ré-exprimé par

$$|\|\mathbf{v}_i - \mathbf{v}_j\| - d_{i,j}| \leq \varepsilon_{ij} \text{ pour } (\mathbf{v}_i, \mathbf{v}_j) \in S. \quad (2.7)$$

Une autre manière de poser le problème nommé iMDGP utilise des intervalles pour borner les distances ; le problème consiste donc à trouver les coordonnées des atomes  $\mathbf{v}_1, \dots, \mathbf{v}_n$  à partir

des bornes inférieure et supérieure  $l_{ij}$  et  $u_{ij}$  de la distance  $d_{ij}$  de telle façon que :

$$l_{ij} \leq \|v_i - v_j\| \leq u_{ij} \text{ pour } (v_i, v_j) \in S. \quad (2.8)$$

Nous pouvons donc différencier plusieurs types de problèmes : (i) le problème des *distances exactes* : les valeurs sont déterminées et l'intégralité des distances entre les sommets est connue. On parle de *graphe rigide* (l'ensemble des transformations sur le graphe sont des isométries, la configuration spatiale du graphe est unique mais sensible aux transformations rigides). (ii) le problème des *distances éparses* : les valeurs sont exactes mais seulement un nombre  $m \leq n \times n$  de distances est connu. Nous parlons dans ce cas de *graphe flexible* (plusieurs sommets peuvent être modifiés spatialement sans violer les contraintes de distance). (iii) le problème des *distances bornées* : les distances sont approximatives ou estimées par un intervalle.

Dans cette thèse nous partons de l'hypothèse que l'ensemble ou une partie des distances est connue et exacte. Dans la partie suivante nous étudions les différentes méthodes existantes et en particulier celles s'appliquant à la problématique d'animation de système articulé.

### 2.3.1 Méthodes de résolution classiques

La résolution de DGP fait l'objet de nombreuses études dans beaucoup de domaines. La diversification des problèmes posés amène à de nombreuses méthodes de résolution. Dans cette section, nous recensons celles qui nous ont semblées intéressantes à appliquer à notre problème. Le lecteur qui s'intéressera plus spécifiquement aux techniques dédiées à la reconstruction de molécule trouvera davantage d'informations dans les travaux de Lavor [103, 127] ou ceux de Jermann [87] concernant la modélisation géométrique par contraintes.

#### 2.3.1.1 Méthodes appliquées aux distances exactes

Nous considérons tout d'abord le cas le plus simple où l'intégralité des distances est connue et exacte. Notons que la méthode décrite ci-après s'applique également pour des problèmes DGP caractérisés par une connaissance partielle des distances (section 2.3.1.2).

**Par une décomposition en valeurs singulières** Le problème de DGP peut être résolu dans un temps polynomial. L'approche classique connue sous le nom de *multidimensional scaling* (CMDS) est une solution efficace qui consiste à utiliser une décomposition en valeurs singulières (SVD) de la matrice d'incidence des distances [152, 153].

En considérant l'ensemble des distances  $d_{ij}$ , où  $i, j = 1, \dots, n$  permettant de retrouver les coordonnées  $v_1, \dots, v_n$  par l'équation  $\|v_i - v_j\| = d_{ij}$  tel que  $i, j = 1, \dots, n$ , nous pouvons écrire que

$$\|v_i\|^2 - 2v_i^T v_j + \|v_j\|^2 = d_{i,j}^2 \text{ pour } i, j = 1, \dots, n. \quad (2.9)$$

En considérant que le problème posé est invariant aux translations et rotations, il est logique de fixer un des sommets à l'origine du monde pour centrer le graphe. On peut donc par exemple définir le dernier sommet par  $\mathbf{v}_n = (0, 0, 0)^T$ . L'équation devient :

$$d_{i,n}^2 - 2\mathbf{v}_i^T \mathbf{v}_j + d_{j,n}^2 = d_{i,j}^2 \text{ pour } i, j = 1, \dots, n-1. \quad (2.10)$$

Nous pouvons maintenant définir la matrice des positions  $\mathbf{V}$  et la matrice métrique  $\mathbf{M}$ , telles que

$$\begin{aligned} \mathbf{V} &= \{\mathbf{v}_{ij} : i = 1, \dots, n-1, j = 1, 2, 3\} \text{ et} \\ \mathbf{M} &= \{(d_{i,n}^2 - d_{i,j}^2 + d_{j,n}^2)/2 : i, j = 1, \dots, n-1\}, \end{aligned} \quad (2.11)$$

$\mathbf{V}\mathbf{V}^T = \mathbf{M}$  et  $\mathbf{M}$  est de rang 3 au maximum. Il est intéressant de remarquer que le problème peut être généralisé dans  $\mathbb{R}^k$  tel que  $\mathbf{V} = \{\mathbf{v}_{ij} : i = 1, \dots, n-1, j = 1, \dots, k\}$  et  $\mathbf{M}$  est donc de rang  $k$ .

L'équation  $\mathbf{V}\mathbf{V}^T = \mathbf{M}$  peut être résolue en utilisant la décomposition en valeurs singulières de  $\mathbf{M}$ . Soit  $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$  la décomposition des valeurs singulières de  $\mathbf{M}$ , où  $\mathbf{U}$  est une matrice orthogonale et  $\mathbf{\Sigma}$  une matrice diagonale avec les valeurs singulières de  $\mathbf{M}$  sur sa diagonale. Nous avons donc également  $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{\Sigma}^{1/2}\mathbf{U}^T$  et donc  $\mathbf{V} = \mathbf{U}\mathbf{\Sigma}^{1/2}$  est la solution de l'équation  $\mathbf{V}\mathbf{V}^T = \mathbf{M}$ . Cette propriété se justifie du fait que  $\mathbf{M}$  est une matrice de Gram. (Pour plus de détails ce point est développé dans les travaux de Sippl [152] et Sit [153]).

Cette méthode spécifique au problème où l'intégralité des distances est connue et exacte est simple et efficace. Elle est par ailleurs une bonne base pour résoudre les autres types de problèmes en se ramenant aux conditions nécessaires à son utilisation, connaître avec exactitude l'ensemble des distances caractérisant le graphe. C'est par exemple le cas d'une des méthodes traitant le problème des *distances bornées* en cherchant à déterminer les bonnes valeurs des distances à partir de distances approximatives.

### 2.3.1.2 Méthodes appliquées aux distances éparses exactes

Dans cette section, nous considérons le problème DGP défini par un ensemble incomplet de distances. Soit  $S$  un sous ensemble de l'ensemble des paires de sommets tel que  $(i, j)$  est dans  $S$  si la distance  $d_{ij}$  entre le sommet  $i$  et le sommet  $j$  est donnée. Le problème est donc de déterminer les coordonnées  $\mathbf{v}_1, \dots, \mathbf{v}_n$  des sommets tel que

$$\|\mathbf{v}_i - \mathbf{v}_j\| = d_{ij} \text{ pour } (i, j) \in S. \quad (2.12)$$

Nous présentons ici plusieurs méthodes dédiées à ce type de problème. Certaines ont pour objectif de se ramener à une résolution similaire à celle présentée dans la section précédente 2.3.1.1. Il s'agit dans ce cas de se ramener à la représentation d'un *graphe rigide*. D'autres ont pour objectif de résoudre le problème simplement avec les contraintes de distance

données, nous parlons de résolution de *graphe flexible* où il existe une infinité de solutions. Notons par ailleurs que la plupart de ces méthodes sont également valables dans le cas où l'ensemble des distances est connu.

**DGP classique appliqué aux distances éparées** L'approche classique consiste à se ramener au problème présenté dans la section précédente où l'intégralité des distances est connue et exacte. Sa résolution repose sur l'utilisation de la matrice de Gram  $\mathbf{G} = \mathbf{V}\mathbf{V}^T$  introduite par l'équation 2.11. Plus précisément, le problème des distances éparées peut être résolu par l'utilisation d'une matrice de Gram incomplète dont il faut déterminer les valeurs manquantes sous la contrainte que le rang de  $\mathbf{G}$  soit égal à la dimension souhaitée (3 pour des coordonnées de  $\mathbf{V}$  dans  $\mathbb{R}^3$ ). Ce nouveau problème NP-complet est connu et appelé *complétion de matrice de rang faible*. Sa résolution fait l'objet de nombreuses études, comme par exemple celles reposant sur les techniques de programmation définie semi positive [27, 94, 143]. Pour plus de détails, une vue d'ensemble des travaux effectués dans ce domaine est donnée dans les articles [39, 53].

**Méthodes numériques itératives** Le problème décrit par l'équation (2.7) peut être également formulé comme un problème d'optimisation globale sous contraintes non linéaires par :

$$\min_{\mathbf{V}=(\mathbf{v}_1, \dots, \mathbf{v}_n)} F(\mathbf{V}). \quad (2.13)$$

avec

$$F(\mathbf{V}) = \sum_{(i,j) \in \mathcal{S}} (\|\mathbf{v}_i - \mathbf{v}_j\|^2 - d_{ij}^2)^2. \quad (2.14)$$

Une façon de trouver la solution à ce problème est d'utiliser une descente de gradient. Plusieurs possibilités existent, comme la méthode connue de Newton-Raphson qui résout itérativement l'équation  $F(\mathbf{V}) = 0$ , avec  $\mathbf{V}$  un vecteur de  $\mathbb{R}^n$  et  $f$  une fonction continue et dérivable. Un exposé approfondi est donné par Bjorck [18]. D'autres méthodes de descente de gradient sont applicables comme par homotopie [123] ou en utilisant l'algorithme de Levenberg-Marquardt. Les méthodes de descente de gradient sont efficaces pour des problèmes avec un ensemble éparse de distances exactes tant que le graphe n'est pas trop grand et avec un bon pré-conditionnement. Par ailleurs il est facile d'ajouter d'autres fonctionnelles dans le problème défini dans l'équation 2.14. Cela se traduit par la possibilité d'ajouter des contraintes géométriques d'autres natures (angles, positions, ...). Par exemple il est possible de positionner le graphe en fixant certains sommets. Toutefois leurs non linéarités amènent un grand nombre de minima locaux. Ce type de méthodes nécessite donc une première initialisation spatiale du graphe pour minimiser les risques de solution locale. Dans les prochaines sections nous présentons d'autres alternatives, ayant pour objectif de limiter les minima locaux, basées sur une même formulation du problème.

**Réduction de graphe** Comme présenté dans la section 2.3.1.2, le problème (2.7) peut être formulé comme un problème d'optimisation sous contrainte global non linéaire :

$$\min_{V=(v_1, \dots, v_n)} \sum_{(i,j) \in \mathcal{S}} (\|v_i - v_j\|^2 - d_{ij}^2)^2. \quad (2.15)$$

Cette fonction est différentiable partout et possède plusieurs minima locaux. La valeur  $V = (v_1, \dots, v_n)$  est solution du problème (2.15) si et seulement si  $V$  est un minimum global et la fonction objectif s'annule.

Comme énoncé précédemment, dans le cas général, le problème de DGP ne possède aucune estimation sur la structure spatiale du graphe. Or la complexité du problème (2.15) associée à sa non linéarité nécessite un pré-conditionnement pour une bonne convergence. L'idée de la réduction de graphe consiste donc à remplacer le problème global formulé par l'équation (2.15) par une séquence de plus petits problèmes. Hendrickson [71] montre en particulier que la structure du graphe peut être décomposée par un algorithme de type *divide-and-conquer*, ce qui réduit fortement la complexité du problème.

Une manière de simplifier le problème est d'exploiter les propriétés du graphe comme par exemple les symétries. L'idée est donc dans un premier temps de réduire le problème à la reconstruction de sous-graphes dont les configurations spatiales sont uniques. Cette étape est résolue par l'équation (2.15). Dans un second temps il s'agit de concaténer les sous-graphes itérativement en commençant à chaque fois par le plus petit. Cette méthode présente l'avantage de ne pas nécessiter l'ensemble complet des distances et est donc plus souple vis-à-vis des approximations.

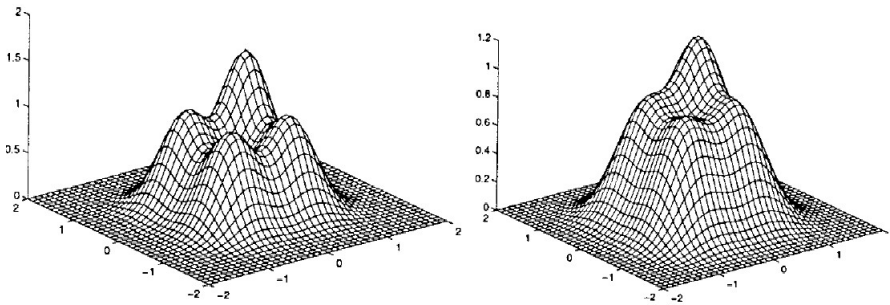


FIGURE 2.1 – Lissage Gaussien par la transformée Gaussienne. La fonction d'origine est telle que  $\lambda = 0$  (à gauche). A droite,  $\lambda = 0.3$  (illustration tirée des travaux de Moré [126]).

**Lissage global et continu** Une autre méthode appelée DGSOL appliquée au problème (2.15) propose d'utiliser un algorithme basé sur du lissage Gaussien. Cette technique comme la précédente est destinée à la résolution de problème avec un ensemble épars de distances.

La méthode introduite par Moré [126] repose sur l'idée que l'on peut trouver le minimum global d'une fonction en la transformant en une fonction plus lisse avec moins de minima lo-

caux. La méthode convertit donc le problème (2.15) en un ensemble de sous fonctions plus simples à étudier. La méthode localise ensuite les minima des fonctions transformées et répercute les changements quand la fonction transformée diffère de l'originale. Une fonction transformée est une approximation grossière de la fonction originale dont le lissage permet d'enlever un grand nombre de minima locaux tout en maintenant sa forme originale. Un exemple de son application est donné par la figure 2.1. Cette fonction  $\langle f \rangle_\lambda$  est appelée transformée de Gauss appliquée à la fonction  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  :

$$\langle f \rangle_\lambda(x) = \frac{1}{\Pi^{n/2}\lambda^n} \int_{\mathbb{R}^n} f(y) \exp\left(-\frac{\|y-x\|^2}{\lambda}\right) dy. \quad (2.16)$$

Dans cette fonction  $\langle f \rangle_\lambda$  est une moyenne de  $f$  au voisinage de  $x$  avec la taille relative du voisinage contrôlée par le paramètre  $\lambda$ . Plus précisément le voisinage de  $f$  décroît en fonction du décroissement de  $\lambda$  et donc si  $\lambda \rightarrow 0$ ,  $\langle f \rangle_\lambda$  converge vers  $f(x)$  (la fonction originale  $f$  est obtenue si  $\lambda = 0$ ). La transformée de Gauss peut également être vue comme la convolution de  $f$  avec une fonction de densité Gaussienne. Elle permet de réduire les hautes fréquences de  $f$ .

Cette méthode est bien plus efficace qu'une simple descente de gradient et permet d'obtenir de bon résultats sans pré-conditionnement pour des graphes de quelques centaines de noeuds (200 environs).

**L'algorithme Build-Up** Développé par Dong et Wu [49], l'algorithme *build-up* propose de résoudre le problème associé à un ensemble de distances exactes dans le cadre de la reconstruction de molécule (MDGP). L'algorithme est basé sur la relation géométrique entre les coordonnées et les distances des atomes de la molécule. L'auteur suppose qu'il est possible de disposer en entrée de l'algorithme d'au moins 4 atomes fixes non coplanaires dont les positions sont connues. La coordonnée  $\mathbf{a}$  d'un atome  $a$  non fixe peut être calculée en utilisant les coordonnées des 4 atomes fixes associés aux distances connues entre les 4 atomes et l'atome  $a$  (figure 2.2). Plus précisément, soit  $b_1, b_2, b_3, b_4$  les 4 atomes fixes dont les coordonnées cartésiennes sont connues. En supposant que les 4 distances Euclidiennes entre l'atome  $a$  et  $b_1, b_2, b_3, b_4$  nommées  $d_{ab_i}$ , pour  $i = 1, 2, 3, 4$  sont également connues, nous avons

$$\|\mathbf{a} - \mathbf{b}_i\| = d_{ab_i}, \quad (2.17)$$

puis

$$\|\mathbf{a}\|^2 - 2\mathbf{a}^T \mathbf{b}_i + \|\mathbf{b}_i\|^2 = d_{ab_i}^2, \quad (2.18)$$

pour  $i = 1, 2, 3, 4$ . En soustrayant l'une de ces équations aux autres, l'auteur obtient un système linéaire qui peut être utilisé pour déterminer les coordonnées de l'atome  $a$ . Par exemple en soustrayant l'équation pour  $i = 1$  aux autres, nous obtenons

$$A\mathbf{x} = \mathbf{b}, \quad (2.19)$$

où

$$A = -2 \begin{bmatrix} (\mathbf{b}_2 - \mathbf{b}_1)^T \\ (\mathbf{b}_3 - \mathbf{b}_2)^T \\ (\mathbf{b}_4 - \mathbf{b}_3)^T \end{bmatrix}, b = \begin{pmatrix} (d_{ab_2}^2 - d_{ab_1}^2) - (\|\mathbf{b}_2\|^2 - \|\mathbf{b}_1\|^2) \\ (d_{ab_3}^2 - d_{ab_2}^2) - (\|\mathbf{b}_3\|^2 - \|\mathbf{b}_2\|^2) \\ (d_{ab_4}^2 - d_{ab_3}^2) - (\|\mathbf{b}_4\|^2 - \|\mathbf{b}_3\|^2) \end{pmatrix} \quad (2.20)$$

Finalement appliqué au système complet, la résolution de ce problème requiert un temps constant puisque le problème doit résoudre  $n - 4$  systèmes pour  $n - 4$  vecteurs de coordonnées  $\mathbf{a}$ . Le temps de calcul est proportionnel à  $n$  tant que 4 distances entre chaque atome et le reste du graphe sont connues.

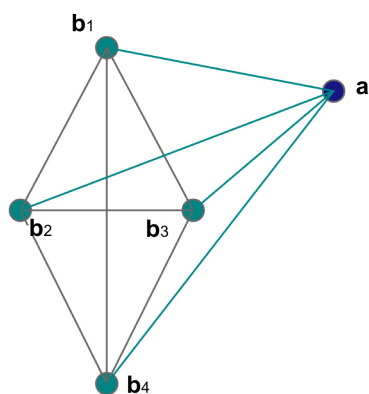


FIGURE 2.2 – L'idée centrale de l'algorithme est de retrouver la position d'un noeud  $a$  à partir de 4 autres noeuds  $b_1, b_2, b_3, b_4$  dont les positions et les distances avec  $a$  sont connues [49].

Cet algorithme a été de nombreuses fois repris et amélioré. Dans cette première version la méthode est sensible aux erreurs numériques liées aux distances. Wu [183] propose donc une amélioration pour prévenir ce problème. Les mêmes auteurs proposent par la suite une adaptation de l'algorithme pour résoudre les cas où le nombre de distances associées à un point est inférieur à 4. Sit [153] propose également une alternative pour prévenir les erreurs numériques en appliquant un algorithme d'optimisation cherchant à approximer la solution par la méthode des moindres carrés.

### 2.3.1.3 Méthodes appliquées aux distances bornées

Dans cette section nous présentons brièvement quelques méthodes destinées à la résolution de problèmes avec des *distances bornées* tels que décrits par l'équation 2.8. Ce type de problème fait l'objet de nombreuses études en Biologie et Chimie pour la reconstruction et la visualisation de molécules.

**L'Algorithme d'Immersion (*Embedding Algorithm*)** Appliqué au problème de reconstruction de molécules à partir de distances bornées, l'algorithme d'immersion [52, 68] se décompose en 3 étapes (1) lissage d'intervalles, (2) correction des métriques, (3) immersion.

A partir d'un ensemble d'intervalles, la première étape (1) consiste à utiliser certaines propriétés géométriques comme l'inégalité triangulaire pour estimer les intervalles de distances

manquants [153]. Une fois ces intervalles estimés, des distances exactes pour chaque paire d'atomes sont générées. Ces distances ne sont pas nécessairement cohérentes entre elles (ne respectent pas l'ensemble des inégalités triangulaires), une seconde étape (2) est alors nécessaire pour corriger les erreurs. Cette procédure se fait par une méthode d'optimisation faisant varier les valeurs des distances entre leurs bornes inférieure et supérieure. Enfin la dernière étape (3) consiste à générer les positions des atomes par une décomposition en valeurs singulières (voir la section 2.3.1.1). Si le rang obtenu est  $\leq 3$ , les distances obtenues lors des deux premières étapes sont satisfaisantes et la solution est correcte. Dans le cas contraire, la solution est approximative et peut être améliorée par une méthode d'optimisation par descente de gradient en considérant les intervalles de distances comme des contraintes (conférer la section 2.3.1.2). L'erreur à mesurer est définie par la fonctionnelle :

$$f_{\varepsilon}(\mathbf{v}_1, \dots, \mathbf{v}_n) = \sum_{(i,j) \in \mathcal{S}} \left( \frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2 - d_{ij}^2}{d_{ij}^2} \right). \quad (2.21)$$

Le problème de cette méthode réside dans la difficulté à identifier les distances erronées ou approximatives.

**Autres méthodes** D'autres méthodes ont été proposées pour résoudre ce type de problème. D'une manière générale, à l'instar de la technique présentée dans la section précédente, la plupart des méthodes existantes sont des variantes des méthodes présentées dans les sections 2.3.1.1 et 2.3.1.2.

Glunt [61] présente une autre version de l'algorithme d'immersion (section 2.3.1.3) en proposant une alternative basée sur une descente de gradient pour ajuster les distances en entrée de chaque itération. Hoai et Dinh développent un algorithme d'optimisation basé sur la différence de fonctions convexes. L'idée est de combiner dans une même résolution le double problème d'optimisation présenté par l'équation (2.15) appliqué aux deux bornes inférieure  $l_{ij}$  et supérieure  $u_{ij}$  qui encadrent la distance  $d_{ij}$ . Les erreurs à minimiser sont équivalentes à celle proposée par l'équation (2.21) appliquée à  $l_{ij}$  et  $u_{ij}$ . La méthode possède toutefois de nombreux minima locaux et nécessite une première initialisation de la configuration du graphe pour être appliquée sur des problèmes plus larges.

## 2.3.2 Applications

Nous listons dans cette section les principales applications des problèmes de DGP.

**Application à la détermination de structures moléculaires (MDGP)** La majeure partie des méthodes proposées ci-dessus sont issues des travaux de recherche concernant la reconstruction de structure moléculaire (figure 2.3). Beaucoup de ces méthodes sont inspirées des travaux de Blumenthal [20] concernant la géométrie des distances. Le nombre de méthodes pour ce type d'application (destiné aux domaines de la Chimie et de la Biologie) s'explique par l'absence de



pré-conditionnement ou de première initialisation de l'état de la molécule. Ainsi les méthodes de type descente de gradient sont difficilement applicables.

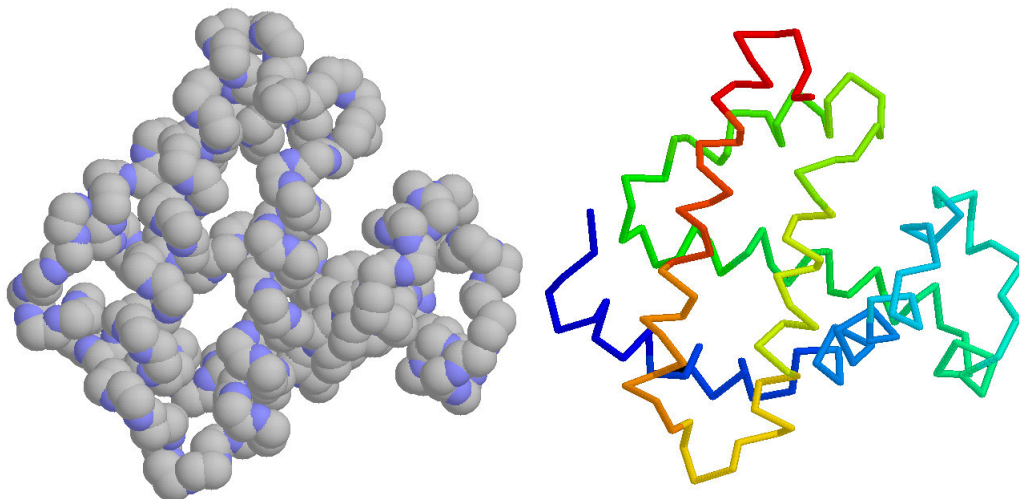


FIGURE 2.3 – Illustrations de la reconstruction d'une molécule [127].

**Application à la modélisation géométrique par contraintes** D'autres études telles que celles proposées par Michelucci [87] s'adressent à des systèmes plus larges considérant plusieurs *objets géométriques* (e.g., points, droites, sphères) associés à plusieurs contraintes géométriques (e.g., positions, distances, angles, alignements). La méthode présentée section 2.3.1.2 répond à ce type de problème. Ces travaux sont davantage destinés à la conception assistée par ordinateur (CAO).

**Application à la reconstruction de réseau de capteurs** Appliqué à la reconstruction de réseaux de capteurs, le problème de DGP est sensiblement différent et est connu sous le nom de *Graph Realization Problem*. Une étude de cas consiste à déterminer la position d'un noeud à partir des distances estimées expérimentalement avec ses voisins les plus proches. L'ensemble des distances caractérisant le problème est donc fortement éparse et les distances approximatives. Ainsi l'objectif est de trouver une représentation planaire des capteurs qui préservent les distances locales. Un tel problème est illustré par la figure 2.4. Le problème posé est clairement non convexe et induit un grand nombre de minima locaux (corrélé à la taille du graphe).

Comme expliqué dans la section 2.3.1.2, plusieurs travaux reformulent le problème comme un Problème de Complétion de la Matrice des Distances Euclidiennes (EDMCP) [27, 94, 143]. l'EDMCP consiste à déterminer les coefficients manquants de la matrice *partielle* de Gram du graphe, et à effectuer une décomposition en valeurs propres (section 2.3.1.1). Ces méthodes utilisent des modèles de programmation définie semi positive SDP, ce qui permet une recherche efficace du minimum global. La résolution par SDP ne garantit toutefois pas le rang de la

solution (3 dans  $\mathbb{R}^3$  et 2 dans le plan) et il est difficile de contraindre la SDP à un rang en particulier.

Certaines méthodes s'appuient sur de l'optimisation globale basée sur une descente de gradient (section 2.3.1.2). Par exemple Weinberger [178] propose de réduire la complexité en utilisant la matrice Laplacienne  $L$  du graphe.  $L$  encode les relations spatiales locales entre les noeuds du graphe. Son étude spectrale permet de fournir un bon préconditionnement à un algorithme type descente de gradient pour reconstituer le graphe.

Les approches *local vers global* découpent le problème original en sous-graphes rigides avant de reconstruire une configuration du graphe entier. Ces techniques enchaînent plusieurs phases comme la *localisation des clusters* (construction de sous-graphes rigides), l'optimisation des clusters (recherche par optimisation type Newton Raphson des positions des noeuds au sein du sous-graphe), transformation des clusters (calcul des rotations et translations à appliquer aux sous-graphes pour les rattacher entre eux).

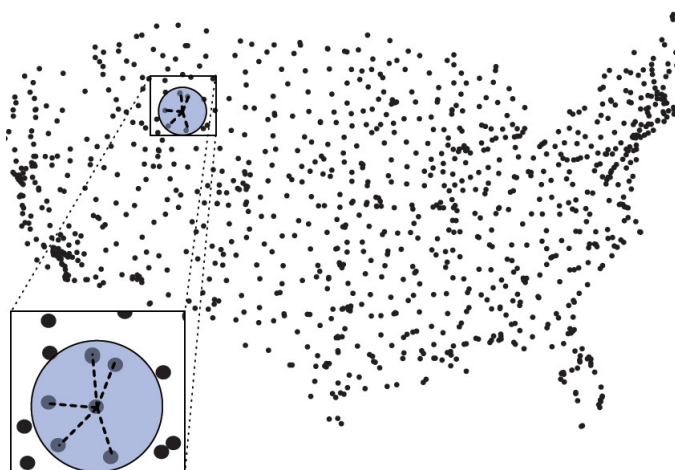


FIGURE 2.4 – Distribution des capteurs sur le territoire des Etats Unis. Les distances sont estimées entre les villes les plus proches. Illustration tirée des travaux de Weinberger [178].

## 2.4 Conclusion et application au mouvement

L'utilisation d'un ensemble complet ou partiel de distances entre sommets pour retrouver leurs positions est un problème connu sous le nom de *géométrie des distances*. Cette problématique se divise en plusieurs sous-problématiques en fonction de la nature des données exploitables et est traitée dans plusieurs domaines. Elle donne donc lieu à plusieurs méthodes de résolution. L'utilisation d'une méthode vis-à-vis d'une autre dépend en particulier de la taille du graphe ainsi que de l'existence d'un pré-conditionnement.

Nous remarquons que les méthodes avec une résolution par SVD sont fiables tant que l'ensemble des distances est complet et les distances sont exactes. Les méthodes de style build-up semblent particulièrement robustes mais ne résolvent pas les problèmes globalement. Les techniques de type descente de gradient sont souples et offrent l'avantage de pouvoir introduire de nouvelles contraintes géométriques autres que les distances. Toutefois elle montrent leurs limites pour la résolution de larges systèmes. Une manière de s'assurer de la bonne convergence est de fournir un bon pré-conditionnement en entrée de l'itération.

**Discussion** Comme nous l'avons présenté dans la première section de ce chapitre (section 2.2), le mouvement dans l'espace métrique est caractérisé par une suite de graphes indépendants les uns des autres dont chaque graphe est associé à un problème de DGP. Nous verrons dans le chapitre suivant qu'à partir des distances extraites d'un mouvement capturé, notre objectif est à la fois de le **rejouer** dans l'espace métrique mais également de le contraindre à de nouvelles tâches pour de la **génération** ou de **l'édition** (modification d'un mouvement existant). Ces tâches introduisent plusieurs problèmes : (i) toutes les distances ne sont pas connues, (ii) l'ajout de nouvelles contraintes de distance peuvent sur-contraindre le problème, (iii) certaines distances ne sont pas exactes. (iv) enfin, ces nouvelles tâches nous amènent à séparer les contraintes en deux sous-ensembles distincts (primaires et secondaires). Les contraintes primaires font référence à la longueur des segments alors que les contraintes secondaires permettent l'édition du mouvement. Ces nouvelles problématiques se formalisent par une définition du problème proche du DGP mais ne permettent pas une exploitation de l'ensemble des méthodes vues précédemment.

Dans le cadre de rejeu de mouvement, où nous disposons de l'ensemble des distances exactes caractérisant le graphe, la méthode classique de DGP (approche spectrale, section 2.3.1.1) est efficace. Elle nécessite toutefois un alignement (translation et rotations) des postures constituant le mouvement a posteriori de la résolution. En revanche, pour de l'édition ou de la génération de mouvement où le problème est sur-contraint et l'ensemble des distances ne sont pas exactes ; cette méthode n'est pas exploitable. Tout comme les problématiques de distances approximatives vues dans la section 2.3.1.3, la résolution de notre système ne donne pas une décomposition en valeurs singulières de rang 3. Cela amène à des approximations perceptibles pour de l'animation (pas de cohérence d'une posture sur l'autre). L'algorithme *Build-Up* décrit dans la section 2.3.1.2 est également applicable pour du rejeu de mouvement, il présente l'avantage d'être rapide et de fournir une résolution exacte en fixant les 4 premiers sommets non coplanaires communs à l'ensemble des postures (pas d'alignement à faire). Cette méthode comme la première est cependant non tolérante aux approximations. Par ailleurs, sa résolution n'est pas globale, l'erreur due à une approximation ou à un problème sur-contraint n'est pas répartie uniformément sur l'ensemble du graphe.

Pour représenter un squelette nous utilisons un ensemble de distances (intra-squelette et distances avec l'environnement) séparé en deux sous-ensembles distincts de contraintes (primaires et secondaires). Naturellement, ces deux ensembles n'ont pas la même importance. Cette pro-

priété nous a motivé à utiliser un processus d'optimisation permettant d'établir des priorités entre les contraintes primaire et secondaire. Nous nous inspirons donc des techniques d'optimisation basées descente de gradient (section 2.3.1.2). Contrairement aux autres approches (par une décomposition en valeurs singulières, build-up, etc.), elles permettent en particulier de projeter les contraintes secondaires dans le noyau de la tâche principale, ce qui constitue un apport intéressant de ces méthodes. Elles permettent également de réduire l'espace des solutions dans le cas où toutes les distances ne sont pas connues.

Dans le chapitre suivant, nous présentons notre méthode d'édition de mouvement dans l'espace métrique et montrons les applications possibles. Nous proposerons également dans le chapitre 4 une méthode de recherche de mouvement dans des bases de données basée sur la même représentation du mouvement dans l'espace métrique.



# 3

## Cinématique guidée par les distances

### Sommaire

---

<b>3.1 Introduction</b>	<b>31</b>
<b>3.2 Approches connexes</b>	<b>32</b>
<b>3.3 Cinématique guidée par les distances : formalisation</b>	<b>34</b>
3.3.1 DGP appliqué à l'animation de personnage	34
3.3.2 Formulation par un problème de minimisation	35
<b>3.4 Méthodes d'optimisation pour résoudre un système d'équations</b>	<b>35</b>
<b>3.5 Adaptation de mouvement et cinématique inverse</b>	<b>36</b>
<b>3.6 Résultats</b>	<b>38</b>
3.6.1 Suivi de mouvement	38
3.6.2 Cinématique inverse	39
3.6.3 Modification de mouvement (cible à atteindre pendant la marche)	42
3.6.4 Application à un graphe cycle	45
<b>3.7 Discussion et perspectives</b>	<b>46</b>

---

### 3.1 Introduction

Traditionnellement, le mouvement d'un personnage est décrit par une suite de squelettes représentés par des vecteurs de positions et rotations. L'animation d'un personnage est générée par la mise à jour dans le temps des valeurs des rotations. Pour éditer ou générer un mouvement, il est classique d'utiliser la méthode de cinématique inverse : à partir d'une pose à atteindre, déterminer les rotations des articulations. Nous pouvons faire un certain nombre de remarques sur la cinématique inverse traditionnelle :

- l'expression d'une tâche se fait par la définition d'un ou plusieurs point(s)  $x$  dans l'espace à atteindre,
- les équations permettant de trouver les positions et les orientations des articulations en fonction du point  $x$  sont hautement non linéaires,

- les systèmes sont généralement redondants vis à vis du problème à résoudre. En effet, le nombre de contraintes est souvent très fortement inférieur au nombre d'inconnues.
- bien que traité et résolu dans la littérature, la cinématique inverse pose le problème de la résolution dans le cas d'effecteurs multiples. Tel que posé, le problème atteint des limites en termes de complexité pour répondre à plusieurs contraintes de cibles. Ainsi la tâche à accomplir est rarement exprimée de manière globale sur l'ensemble du squelette mais plutôt introduite par un ensemble de tâches entre lesquelles il faut émettre des priorités.

Dans le chapitre précédent, nous avons présenté le problème général de l'application du problème de géométrie des distances à l'animation de squelette. Nous avons dans ce contexte expliqué quelle représentation du mouvement nous avons choisie (section 2.2) et dans quel intérêt une telle représentation est pertinente. Nous avons également décrit le problème fondamental de la Géométrie des distances (section 2.3) et quelles méthodes existent pour le résoudre.

**Plan du chapitre.** Après avoir décrit les approches connexes, nous formalisons le problème de la cinématique à partir de l'expression de distances (section 3). Dans ce cadre, un squelette est caractérisé par un ensemble de points, chaque point s'exprimant par rapport aux autres en termes de distances. Nous proposons une méthode pour résoudre le problème de cinématique directe et mettons en évidence comment à travers cette formulation spatiale du squelette nous pouvons traiter le cas de la cinématique inverse (sections 4 et 5). Enfin, nous proposons plusieurs expérimentations afin de valider notre méthode (section 6).

## 3.2 Approches connexes

**Cinématique inverse et applications** Les techniques de cinématique inverse ont été largement étudiées au cours des dernières décennies, dans un contexte à la fois graphique [179, 191] mais surtout robotique [131, 182]. Ces méthodes restent un outil de choix, à la fois pour générer des mouvements a priori inconnus, mais aussi comme composant de méthodes plus spécifiques comme l'adaptation de mouvements à de nouvelles contraintes [28, 38, 59, 69, 101, 125], l'inférence de poses basée sur des capteurs visuels [67, 86], ou bien encore la reconstruction de mouvements partiellement acquis depuis des méthodes de capture [7, 63]. Les approches les plus connues ont fait l'objet d'études comparatives dans le cadre de la reconstruction de différentes classes de mouvement [174].

Cependant, le mauvais conditionnement mathématique du problème, couplé à l'importante redondance des chaînes articulées considérées (que ce soit pour un bras ou une jambe, mais à plus forte raison pour un humanoïde complet), nécessite l'adjonction de tâches secondaires qui viennent sur-contraindre le problème [12, 24, 147, 187]. Notons comme applications possibles de ces tâches secondaires la gestion des butées articulaires [42], ou bien encore le maintien de l'équilibre d'une posture [24]. Des travaux plus récents s'intéressent à l'intrication de plusieurs niveaux de contraintes [12, 28, 147] à l'aide de niveaux de priorité. La difficulté est ici de faire coexister des tâches potentiellement antagonistes sans idée a priori sur la réalisation globale de

l'ensemble des consignes.

Du point de vue de la résolution du problème inverse, dans la grande majorité des cas, les algorithmes existants sont basés sur une descente de gradient exploitant une inversion de la Jacobienne évaluée numériquement. L'une des méthodes les plus efficaces consiste à calculer la pseudo-inverse de Moore-Penrose associée à une fonction d'amortissement [114]. Cependant, il existe plusieurs alternatives à ce fonctionnement : approches sensori motrices [57], approches probabilistes [63] ou inférence Bayésienne [41]. Notons aussi la possibilité d'utiliser des informations issues de mouvements capturés comme poses de référence dans les processus d'optimisation [31, 63, 140, 173]. Ces approches travaillent non plus directement dans l'espace des poses mais plutôt dans des espaces latents (obtenus par méthodes de type analyse en composantes principales), sur-contraignant ainsi le problème et traduisant de manière implicite les corrélations pouvant exister entre certaines articulations. Notons ici que dans la méthode que nous proposons, l'utilisation de poses de référence, définies comme un ensemble de distances à respecter, joue un rôle similaire à ces approches basées données.

D'autres méthodes proposent de résoudre le problème de cinématique inverse en se plaçant, non plus dans l'espace des orientations mais dans celui des positions (espace euclidien). Ainsi, la méthode FABRIK [7] propose un nouvel algorithme de cinématique inverse qui permet de remonter de proche en proche aux différentes positions des articulations d'une chaîne, en parcourant la chaîne en avant et en arrière en un nombre fini d'itérations. L'intérêt principal de cette approche réside dans la possibilité de contrôler des systèmes multi-effecteurs mais également dans sa rapidité d'exécution. Cette méthode, qui s'exprime sous la forme d'une séquence de constructions géométriques locales, ne garantit pas cependant l'optimalité de la solution obtenue. Dans la méthode intitulée *Particle IK Solver* [69], le personnage est représenté par un ensemble de particules 3D (positions des articulations). La méthode proposée se ramène à une méthode de résolution itérative sous contraintes, qui s'appuie sur un ensemble de préconditions ad hoc et opère en deux phases distinctes appliquées successivement aux différentes parties du corps à animer. Cette décomposition du problème, en permettant un contrôle local, apporte des solutions flexibles et paramétrables qui préservent le style des mouvements. Ho et co-auteurs [76] s'intéressent également à une représentation du squelette et du maillage sous la forme d'un nuage de points. Mais contrairement aux deux approches précédentes, le problème d'adaptation du mouvement (en particulier l'adaptation morphologique) est formalisé comme un problème global d'optimisation spatio-temporelle, les variations de position étant calculées à partir de la minimisation d'une somme de grandeurs énergétiques (déformation, accélération, contrainte énergétique). Enfin, une nouvelle méthode de résolution d'IK, basée sur des contraintes de distance est proposée en robotique [66]. Cette méthode géométrique consiste à paramétrer le système par des distances inter-articulations, et à reformuler le problème à partir de la résolution d'un système d'inégalités linéaires.

**Application de la DGP à l'animation de squelette** De manière similaire aux approches précédentes, notre méthode représente le personnage à animer à partir d'un ensemble de po-



sitions cartésiennes. Elle permet, comme dans [7] et [66] de gérer des effecteurs multiples et des graphes cycles. Cependant, notre méthode diffère essentiellement de la méthode de résolution de l'IK. Pour se faire nous étendons le Problème de Géométrie des Distances (DGP) introduit dans le chapitre 2 et nous nous inspirons des techniques d'optimisation basées descente de gradient (section 2.3.1.2). La justification de ce choix est par ailleurs développé dans la section 2.4.

### 3.3 Cinématique guidée par les distances : formalisation

#### 3.3.1 DGP appliqué à l'animation de personnage

Notre étude porte sur l'animation d'un squelette dans l'espace des distances. Chaque articulation peut être représentée géométriquement par un point dans  $\mathbb{R}^3$ . En terme de distance, nous cherchons à exprimer l'écart d'une articulation avec d'autres articulations du squelette ou encore avec des points de l'espace. Nous reprenons ici les concepts définis dans la section 2.3 du chapitre 2 appliqué au squelette. Rappelons que nous représentons le squelette par le graphe  $G = (\mathbf{V}, \mathbf{E}, d)$  où  $\mathbf{V}$  est l'ensemble des points (centres des articulations),  $\mathbf{E}$  l'ensemble des arêtes reliant les points de  $\mathbf{V}$  et  $d$  l'ensemble des distances de  $\mathbf{V} \times \mathbf{V} \rightarrow \mathbb{R}_+$ . Nous noterons en particulier  $d_{ij}$  la distance euclidienne entre le point  $\mathbf{x}_i$  et le point  $\mathbf{x}_j$ , pour tout  $(\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{E}$ . Nous pouvons exprimer la structure spatiale de ce graphe par l'ensemble des équations :

$$\forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{E}, \|\mathbf{x}_i - \mathbf{x}_j\| = d_{ij}, \quad (3.1)$$

où  $\|\cdot\|$  est la norme euclidienne. Pour faciliter l'écriture, nous utilisons indifféremment la notation  $\mathbf{x}_i$  pour représenter le point de  $\mathbf{V}$  ou le vecteur  $\overrightarrow{O\mathbf{x}_i}$ .

Notre problème peut se formuler de la manière suivante : à partir d'un ensemble suffisant de distances, il s'agit de retrouver les positions des points (et donc des articulations) dans l'espace  $\mathbb{R}^3$ .

Chaque point de  $\mathbf{V}$  est potentiellement associé à un certain nombre de distances dépendant de la topologie du graphe en ce point. Remarquons dès à présent que pour inférer un point  $\mathbf{x}_i$  dans l'espace il suffit de disposer de quatre points connus et deux à deux distincts, associés à leurs distances respectives au point  $\mathbf{x}_i$ . Ceci implique que suivant les cas nous pouvons simplifier le problème posé dans l'équation (3.1).

Numériquement, résoudre de telles équations s'avère difficile. Il existe plusieurs méthodes. Les méthodes par approche spectrale répondent bien au problème mais ne s'appliquent toutefois qu'à des situations où toutes les distances sont connues et exactes (respect des inégalités triangulaires). Nous avons choisi d'utiliser des méthodes d'optimisation qui permettent en particulier d'approcher les solutions dans le cas où les distances ne sont pas exactes, ou encore de réduire l'espace des solutions dans le cas où toutes les distances ne sont pas connues.

### 3.3.2 Formulation par un problème de minimisation

Nous étendons ici les concepts vus dans la section 2.3.1.2 dédiée à la résolution de problème à distances éparse par des techniques d'optimisation. Résoudre le problème en passant par des méthodes d'optimisation revient à reformuler l'équation (3.1) sous la forme d'un problème de minimisation d'énergie  $W$  tel que :

$$W = \arg \min_{\mathbf{X}} \sum_E (\|\mathbf{x}_i - \mathbf{x}_j\|^2 - d_{ij}^2)^2, \quad (3.2)$$

où  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$  est le vecteur d'inconnues à déterminer, avec  $n = |\mathbf{V}|$  et  $m = |E|$ .

Remarquons que dans l'équation (3.2), nous effectuons une pénalisation quadratique de l'erreur. L'intérêt est d'assurer une erreur positive pour chaque équation, ce qui implique que plusieurs erreurs ne peuvent pas mutuellement s'annuler. Il en résulte une meilleure stabilité de l'algorithme.

Nous remarquons également que l'équation (3.2) est un système d'équations non linéaires.

Nous appellerons dans la suite de l'article *contraintes* les règles que doivent satisfaire les solutions réalisables du problème. Parmi celles-ci l'égalité de l'équation (3.1) (contrainte de distance) ou encore l'égalité d'un point avec un autre ( $g(\mathbf{x}) = \mathbf{x}_T$ ) caractérisent les contraintes que nous pouvons donner à notre système. Appliqué à un squelette, l'objectif du problème revient alors à associer à chaque articulation un nombre suffisant de contraintes.

Afin de décrire plus précisément la méthode d'optimisation, nous traiterons uniquement dans la section suivante des contraintes de distance établies dans l'ensemble des points  $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ . L'ensemble de ces contraintes peut être mis sous la forme d'un système d'équations  $F$  de  $m.n$  fonctions objectifs  $f_{ij}$  où chaque fonction reprend l'égalité de distance entre deux points  $\mathbf{x}_i$  et  $\mathbf{x}_j$  telle que :

$$f_{ij}(\mathbf{x}_i, \mathbf{x}_j) = (\|\mathbf{x}_i - \mathbf{x}_j\|^2 - d_{ij}^2)^2 = 0. \quad (3.3)$$

Dans ce cadre, le problème revient donc à résoudre le système d'équations  $F(\mathbf{X}) = 0$ , où  $F = (f_{11}, \dots, f_{1n}, \dots, f_{mn})^T$  représente le vecteur des fonctions objectifs et  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$  est le vecteur des variables inconnues.

## 3.4 Méthodes d'optimisation pour résoudre un système d'équations

Le système d'équations étant non linéaire, une approche par descente de gradient telle que la méthode de Newton est utilisée. Elle présente l'avantage d'être rapide et suffisamment stable pour notre problème. En effet une animation supposant une continuité temporelle, cela introduit une faible variation des postures du squelette entre chaque itération de l'algorithme. La formule

de Newton appliquée à notre problème s'écrit :

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{J}_F(\mathbf{X}_k)^{-1} F(\mathbf{X}_k), \quad (3.4)$$

où  $\mathbf{X}_k$  est le vecteur  $\mathbf{X}$  obtenu à l'itération  $k$  de l'optimisation et  $\mathbf{J}_F(\mathbf{X}_k)$  est la matrice jacobienne de  $F(\mathbf{X}_k)$  au point  $\mathbf{X}_k$ .

La matrice jacobienne  $\mathbf{J}$  rassemble les dérivées partielles de toutes les contraintes du problème  $F$ , c'est une matrice de dimension  $m \times n$  ( $m \times n$  contraintes pour  $n \times 3$  inconnues) :

$$\mathbf{J}(\mathbf{x}_1, \dots, \mathbf{x}_n) = \begin{pmatrix} \frac{\partial f_{11}}{\partial \mathbf{x}_1} & \cdots & \frac{\partial f_{1n}}{\partial \mathbf{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{m1}}{\partial \mathbf{x}_1} & \cdots & \frac{\partial f_{mn}}{\partial \mathbf{x}_n} \end{pmatrix}. \quad (3.5)$$

$$\text{avec } \frac{\partial f_{ij}}{\partial \mathbf{x}_j} = \left( \frac{\partial f_{ij}}{\partial x_j} \frac{\partial f_{ij}}{\partial y_j} \frac{\partial f_{ij}}{\partial z_j} \right)^T$$

$x_j, y_j$  et  $z_j$  étant les coordonnées du point  $\mathbf{x}_j$ .

Nous remarquons que le problème précédemment posé est plus général et qu'il n'amène pas forcément à un nombre de fonctions égal au nombre de variables inconnues. En effet, en fonction du nombre de contraintes définies par l'utilisateur, nous pouvons nous ramener à des problèmes sous-déterminés similaires à ceux de la cinématique inverse standard, avec plus d'inconnues que de contraintes à respecter. A contrario, notre méthode permet simplement d'ajouter des contraintes de distance réduisant l'espace de solution, jusqu'à se retrouver dans des problèmes sur-contraints (plus de contraintes que d'inconnues). Mathématiquement, ces différentes situations conduisent à des matrices jacobiennes qui ne sont pas forcément carrées, ni de rang plein. Nous utilisons donc la matrice pseudo-inverse (ou inverse de Moore-Penrose)  $\mathbf{J}^+$  qui est une généralisation de l'inverse caractérisant toute matrice  $m \times n$  ( $m > n$ ) telle que  $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ . Le calcul de la pseudo-inverse est réalisé par la méthode des valeurs singulières (SVD), et l'évaluation de la matrice jacobienne est effectuée à travers l'expression analytique du gradient de la contrainte de distance entre deux points. Ainsi pour un élément de la matrice  $\mathbf{J}$  évalué aux points  $\mathbf{x}_i$  et  $\mathbf{x}_j$  dans  $\mathbb{R}^3$ , le gradient s'écrit :

$$\frac{\partial f_{ij}}{\partial \mathbf{x}_j}(\mathbf{x}_i, \mathbf{x}_j) = -4 \cdot (\mathbf{x}_i - \mathbf{x}_j) \cdot f_{ij}(\mathbf{x}_i, \mathbf{x}_j) \quad (3.6)$$

### 3.5 Adaptation de mouvement et cinématique inverse

Pour résoudre des problèmes de cinématique inverse pure, le modèle décrit ci-dessus est pleinement satisfaisant. Si l'on considère une chaîne articulée fixée à une extrémité, connaissant la posture initiale et une cible à atteindre par l'autre extrémité, il suffit en effet de donner au système les contraintes de distance entre les articulations adjacentes du squelette.

Le problème ainsi formulé nous permet de rejouer un mouvement ou encore de résoudre la cinématique inverse en minimisant itérativement à chaque posture clé de l'animation les différentes contraintes de distance. Dans l'objectif d'adapter un mouvement, il est nécessaire d'introduire de nouveaux concepts et d'améliorer la méthode d'optimisation précédente. Nous pouvons différencier deux types de contraintes de distance (figure 3.1) :

- **Contraintes dures** : ce sont les contraintes qui doivent être impérativement respectées. Il s'agit par exemple des distances invariantes dans le temps qui sont les distances entre les articulations adjacentes dans la hiérarchie du squelette. Elles correspondent aux segments. Il s'agit également des contraintes cibles de la cinématique inverse, comme par exemple le contact d'un pied avec le sol représenté par une distance nulle entre le pied et un plan. Pour ces contraintes les fonctions doivent être nulles à l'issue de l'itération de la boucle d'optimisation.
- **Contraintes souples** : ce sont les autres distances du graphe. Dans le cadre de la cinématique directe, elles évoluent dans le temps et ce sont finalement elles qui caractérisent l'animation.

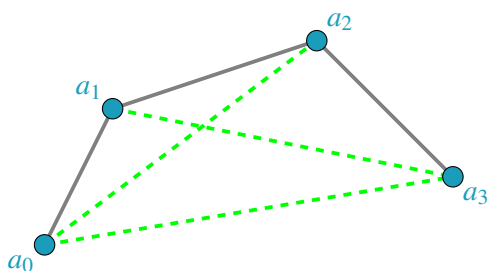


FIGURE 3.1 – Les segments en pointillés vert représentent les contraintes de distance souples alors que les autres segments représentent les contraintes de distance dures.

L'introduction de ces contraintes est motivée par la nécessité de donner une priorité à certaines contraintes par rapport à d'autres. Dans le cadre de l'adaptation de mouvement, nous partons d'un mouvement référence calculé dans l'espace des distances auquel on ajoute une tâche via des contraintes dures. Ceci induit un conflit entre les contraintes : la longueur des segments ainsi que la tâche ajoutée sont des contraintes à satisfaire en priorité (contraintes dures), alors que les distances qui caractérisent le positionnement spatial des articulations de la posture de référence constituent des propriétés que l'on souhaite conserver au mieux (contraintes souples). Le système ainsi défini est finalement sur-contraint. Nous proposons donc une modification de l'algorithme de Newton qui consiste à ajouter une tâche secondaire (contraintes souples) qui est projetée dans l'espace des solutions de la tâche principale (contraintes dures). Cela permet d'incorporer les contraintes dures et souples dans un même algorithme en tenant compte éventuellement des priorités entre les contraintes, comme c'est le cas dans la formulation proposée

par Boulic co-auteurs [28]. Notre algorithme devient :

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{J}_0^+ F(\mathbf{X}_k) + (\mathbf{I} - \mathbf{J}_0^+ \mathbf{J}_0) \mathbf{J}_1^+ G(\mathbf{X}_k) \quad (3.7)$$

où  $\mathbf{X}_k$  est le vecteur des positions des articulations à déterminer. Par souci de simplification des notations,  $\mathbf{J}_{F_0}^+(\mathbf{X}_k)$  et  $\mathbf{J}_{F_1}^+(\mathbf{X}_k)$  sont notées  $\mathbf{J}_0^+$  et  $\mathbf{J}_1^+$ .  $F(\mathbf{X}_k)$  et  $G(\mathbf{X}_k)$  représentent respectivement les fonctions objectifs associées aux contraintes dures et souples, et  $\mathbf{J}_0^+$  et  $\mathbf{J}_1^+$  sont les pseudo-inverses des jacobiniennes associées.  $(\mathbf{I} - \mathbf{J}_0^+ \mathbf{J}_0)$  est un opérateur de projection qui garantit que la réalisation de  $G(\mathbf{X}_k)$  n'affectera pas celle de  $F(\mathbf{X}_k)$ .

## 3.6 Résultats

Afin de valider notre méthode, nous l'expérimentons sur plusieurs cas d'études. Nous mettons ainsi en évidence la convergence de la méthode et l'importance de la définition des tâches à accomplir. Tout d'abord nous nous intéressons à la tâche de suivi de mouvement qui a pour objectif de rejouer une animation. Dans un second temps, nous expérimentons des situations de cinématique inverse, notamment en comparant les résultats à ceux obtenus par une méthode de cinématique traditionnelle. Enfin nous testons notre méthode sur une expérience d'adaptation de mouvement. Nos tests sont tous réalisés en temps réels (10 images par seconde pour l'animation la moins rapide). Ils sont effectués par un seul processus avec un processeur cadencé à 2.3 GHz. Nous avons testé et utilisé deux bibliothèques mathématique ([46], [64]) pour valider nos résultats.

### 3.6.1 Suivi de mouvement

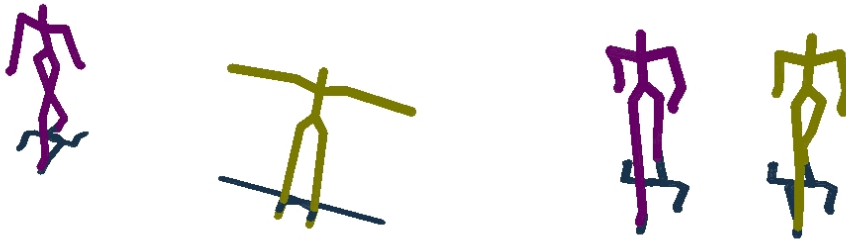


FIGURE 3.2 – Illustration de la reproduction de mouvement par tracking. Figure de gauche : posture de référence (à gauche), posture initiale (à droite). Figure de droite : le mouvement de référence (à gauche) et le mouvement calculé (à droite).

De façon à valider la capacité de notre méthode à calculer une animation dans l'espace des distances, notre premier test porte sur un exemple rejoué, qui représente la reproduction d'un mouvement source  $S$  (données capturées) vers un mouvement cible  $C$  pour des squelettes

identiques. Dans ce contexte, nous définissons quatre points deux à deux distincts dans l'espace (soit la base canonique et l'origine). Pour chaque pas de temps, après avoir calculé les positions des articulations de  $S$ , nous extrayons pour chaque articulation du squelette  $S$  les distances avec ces quatre points. Enfin nous incorporons ces distances dans notre système. Plus précisément, pour un squelette à  $n$  articulations, nous définissons un problème de  $4n$  contraintes pour  $3n$  inconnues. Notre test est effectué sur un mouvement de marche stocké au format *bvh* provenant de la base CMU.

Afin de valider notre méthode, nous évaluons la convergence de l'algorithme, et nous montrons (figure 3.2) le suivi en temps réel de l'animation cible. La figure 3.3 montre bien le succès de la convergence. Plus précisément, nous pouvons en déduire que la minimisation des erreurs pour chaque contrainte à l'issue de 10 itérations est égale à zéro (rappelons que la minimisation a été élevée au carré). Cela se traduit par une distance nulle entre les articulations de  $S$  et  $C$ , et visuellement par un suivi à l'identique de l'animation  $C$  sur  $S$  (figure 3.2).

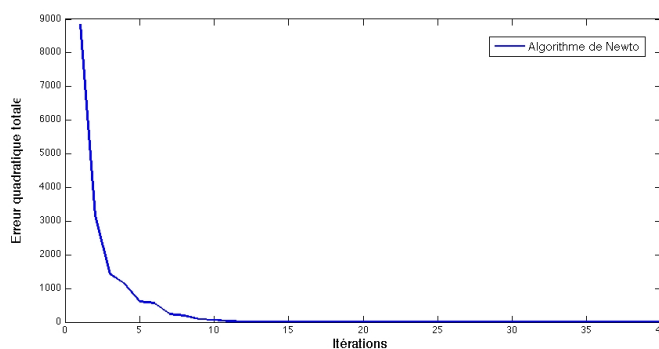


FIGURE 3.3 – Résultats de convergence par l'itération de Newton (passage de la pose de référence à la première pose du mouvement)

Cette expérimentation montre également la robustesse de la méthode : en effet, comme l'illustre la figure 3.2, l'algorithme d'optimisation peut s'adapter à des variations importantes de valeurs initiales (posture très différente de la posture de référence). L'animation de personnage présente l'avantage d'avoir une pause de référence (bind pose) qui introduit un vecteur initial dans l'itération de Newton déjà très proche de la solution.

## 3.6.2 Cinématique inverse

### 3.6.2.1 Atteinte d'une cible par une chaîne articulée

Notre second test (figure 3.4) porte sur un exemple simple de cinématique inverse. L'objectif est de faire atteindre une cible  $c$  à l'extrémité d'une chaîne articulée  $e_1$ , l'autre extrémité  $e_2$  étant attachée à un autre point  $p$ . Pour cette expérience, nous construisons une chaîne articulée formée de 10 articulations, chacune espacée de la même distance de l'articulation adjacente. Le

problème se pose ainsi : nous donnons au système les 9 contraintes propres à la chaîne articulée plus 2 contraintes, de telle façon que les distances entre  $e_1$  et  $c$  et entre  $e_2$  et  $p$  soient nulles. Il s'agit ici de mettre en évidence le bon respect des contraintes (11 contraintes pour  $3 \times 10$  inconnues).

Les résultats montrent en premier lieu le succès de l'expérience avec des contraintes respec-

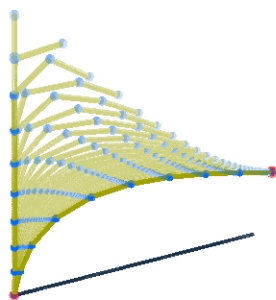


FIGURE 3.4 – Illustration du test du bras articulé. Les contraintes ajoutées sont illustrées par les sphères rouges.

tées (l'erreur est à zéro après 18 itérations). Au delà du respect de ces contraintes, l'expérience montre que notre méthode est capable sur un exemple simple de réaliser les même tâches que celles résolues par la cinématique inverse traditionnelle.

Notre méthode ainsi résolue présente cependant l'inconvénient d'être moins rapide que les méthodes standard de cinématique inverse. En effet, comme illustré dans la figure 3.5, nous avons comparé les temps moyens de calcul d'une itération en fonction du nombre d'articulations d'une chaîne pour notre méthode et pour la cinématique inverse standard. Les différences de temps entre les méthodes s'expliquent par le fait que notre matrice jacobienne basée distance est dans le cadre de cette implémentation de dimension bien supérieure à la matrice jacobienne classique (basée orientation), conduisant par conséquent à un temps de calcul de l'inversion plus important. Remarquons toutefois que notre matrice est fortement creuse et que l'on peut envisager, pour accélérer l'inversion de ce type de matrice, des méthodes telles que celle proposée par M. Berry [16]. L'accélération de la méthode dans ce contexte est implémentée et discutée dans la section 6.3. Notons également quelques irrégularités dans la courbe basée distance, dues à des variations de la charge du processeur.

Dans le cas de la cinématique inverse standard, les problèmes de singularité amènent à utiliser une matrice d'amortissement. Les résultats obtenus sont fortement dépendants de la bonne estimation des coefficients de cette matrice. Il est intéressant de noter que notre méthode donne une configuration régulière de la chaîne sans paramètre d'amortissement (voir figure 3.6). Cela s'explique par le fait que la minimisation est calculée dans l'espace des distances, ce qui est équivalent à exprimer les variations à travers la jacobienne le long de l'axe entre deux points

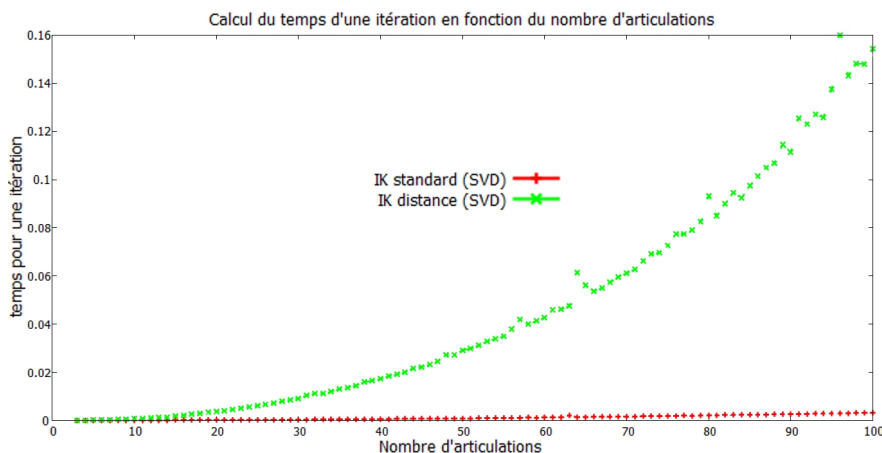


FIGURE 3.5 – Temps moyen de calcul d’une itération en fonction du nombre d’articulations pour les méthodes basées distance (en vert) et les méthodes de cinématique inverse traditionnelle (en rouge).

contraints, contrairement à la cinématique inverse standard qui applique ces variations sur les 3 axes de rotation de chaque articulation.

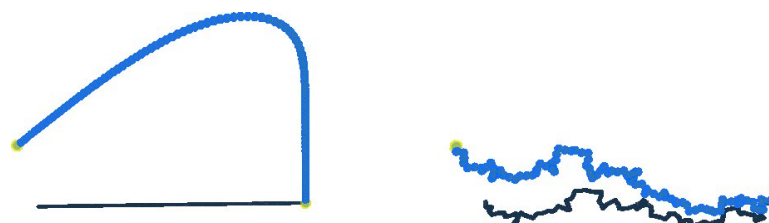


FIGURE 3.6 – Illustration pour deux chaînes identiques. A droite : calculé par la cinématique inverse standard avec un mauvais facteur d’amortissement. A gauche : calculé par notre méthode.

### 3.6.2.2 Application au contact d’une main avec un objet

A travers cette expérience, nous cherchons à mettre en évidence la capacité de la méthode à exprimer simplement des tâches sous la forme de distances, ainsi qu’à gérer implicitement un nombre important de contraintes associées à un certain niveau de priorité. Cette application montre également l’intérêt d’exprimer une articulation vis-à-vis des autres du système. Cela permet en particulier de préserver une configuration spatiale cohérente lors de la génération du mouvement.

Dans le cadre de la cinématique inverse traditionnelle, il est possible également de spécifier des tâches complexes sous la forme de buts multiples à atteindre, caractérisés par des contraintes géométriques de différentes sortes [138]. Ce type d’approche peut se décomposer



en deux phases :

- il s’agit d’abord de convertir les contraintes en termes de buts à atteindre,
- puis de résoudre la cinématique inverse dans l’espace des coordonnées angulaires.

Notre méthode s’affranchit de la phase de conversion. Elle permet de spécifier directement la tâche sous la forme d’un ensemble de propriétés à respecter. Ces propriétés sont implicitement intégrées dans la matrice jacobienne, et l’algorithme d’optimisation cherche à trouver une solution convergente dans l’espace des coordonnées cartésiennes.

Afin d’illustrer cette approche, nous avons mis en oeuvre une expérience (voir figure 3.7) d’atteinte d’une sphère par la main droite, à partir de la simple spécification d’un ensemble de distances, sans connaître précisément les points à atteindre. L’expérience est réalisée avec un squelette détaillé de la main. A partir d’une posture clé de référence, l’objectif est d’amener l’ensemble des articulations de la main droite à toucher la sphère placée dans la scène 3D. D’un côté, les contraintes souples correspondent aux autres distances permettant d’exprimer une articulation vis-à-vis des autres (par exemple l’articulation d’un doigt vis-à-vis du reste de la main). De l’autre, au delà des longueurs des segments, nous avons considéré des contraintes dures qui expriment la tâche à travers les contacts entre les 26 articulations de la main et la sphère. Ainsi, toute articulation de la main en contact avec la sphère conduit à une contrainte de distance qui s’exprime sous la forme d’une fonction objectif du même type que l’équation (3.3).

Cet exemple montre l’intérêt d’incorporer des priorités entre les contraintes, sous la forme de contraintes dures et de contraintes souples (tâches secondaires). Ne pas spécifier de priorité surcontraint le problème, avec le risque qu’il n’y ait pas de solution. Il est possible ainsi, en ajoutant ou enlevant des contraintes, d’influer sur les solutions du problème d’inversion. Si l’on enlève par exemple la tâche secondaire, cela donne plus de latitude au système d’atteindre la cible, mais peut conduire à une posture de la main potentiellement irréaliste.

### 3.6.3 Modification de mouvement (cible à atteindre pendant la marche)

Nous mettons ici en évidence la capacité de la méthode à modifier des mouvements d’origine, en introduisant des tâches supplémentaires. Nous montrons également deux phases dans le déroulement de l’animation. La première concerne l’atteinte de l’objectif (positionnement d’une contrainte), la seconde le relâchement de la contrainte qui s’accompagne du retour du squelette vers la posture initiale. Cette expérimentation soulève le problème de la concurrence entre les tâches, ce qui nécessite une gestion de la priorité entre contraintes.

Pour cet exemple, nous reprenons le mouvement de marche de l’expérience de suivi de mouvement. Durant sa marche, nous donnons comme directive au personnage d’atteindre successivement par la main droite, la main gauche puis à nouveau la droite des cibles qui sont sur son itinéraire. En reprenant les mêmes contraintes que celles de la première expérience, nous considérons cette fois-ci les longueurs des segments comme des contraintes dures et les autres comme des contraintes souples (tâche secondaire). L’objectif d’atteindre des cibles du-

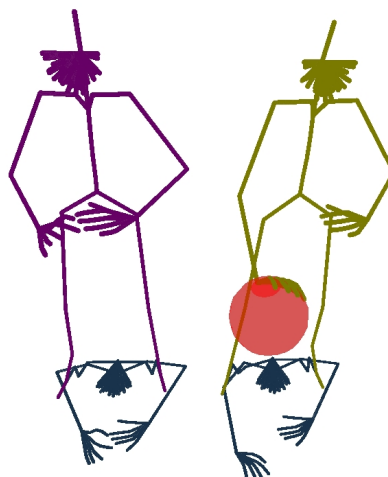


FIGURE 3.7 – Sphère à atteindre avec la main droite. A gauche la pose de référence. A droite le mouvement composé.

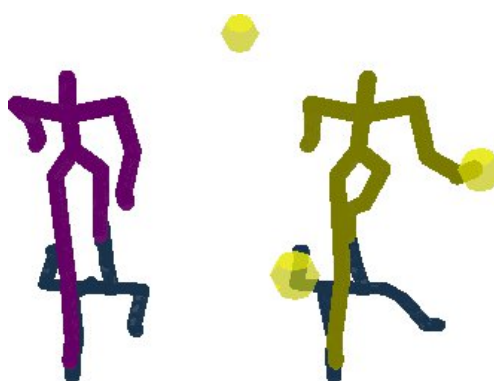


FIGURE 3.8 – Illustration de l'adaptation de mouvement. A droite le mouvement référence, à gauche le nouveau mouvement.

rant l'animation se traduit par l'ajout temporaire de contraintes dures prioritaires sur la tâche secondaire.

La figure 3.9 met en évidence le conflit entre les contraintes à travers une augmentation de l'erreur de la tâche secondaire lors de l'atteinte des cibles et le bon respect des contraintes dures tout au long de l'animation.

**Stabilité** Dans un premier temps, il est important de noter que, contrairement à la cinématique inverse standard où l'orientation d'une articulation est dépendante de l'ensemble des articulations parentes, travailler sur un nuage de points permet d'exprimer un point par rapport à deux voisins dans le cas le plus simple et rarement à plus de quatre ou cinq points pour un

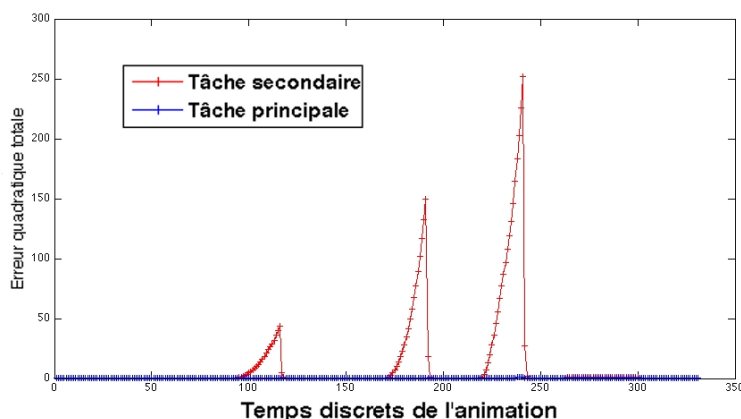


FIGURE 3.9 – Évolution des erreurs des tâches principale et secondaire au cours de l'animation. En bleu, l'erreur générée par la tâche principale. En rouge, celle pour la tâche secondaire.

squelette (six pour un maillage régulier). Cela implique une très faible dépendance entre les points et donc une répartition plus locale et plus spécifique à la fonction objectif de l'erreur, ce qui rend notre algorithme plus stable. Notons cependant que cette méthode conduit à une convergence plus lente. Dans un second temps, le fait de sur-contraindre le problème entraîne la possibilité d'absence de solution et peut donc conduire à une non continuité entre deux itérations. Notons enfin que nous avons ajouté un paramètre à l'itération de Newton visant à limiter l'implication de la tâche secondaire en fin d'itération, la tâche secondaire servant ainsi de guide vers une solution durant l'itération.

**Performance** Dans ce contexte, une manière d'optimiser le problème consiste à exprimer l'ensemble des contraintes souples vis-à-vis de quatre points de référence statiques (qui ne font pas partie du vecteur d'inconnues), ce qui signifie également qu'il n'y a pas de dépendance entre les trois coordonnées d'un point avec toutes les autres coordonnées du vecteur d'inconnues. Cela induit que notre matrice jacobienne des contraintes souples est une matrice diagonale par blocs. Or l'inverse d'une telle matrice est une matrice diagonale par blocs des inverses des blocs. On peut donc simplifier et finalement accélérer la vitesse de l'inversion de notre jacobienne en calculant indépendamment les pseudo-inverses de chaque sous matrice. Si  $\mathbf{x}_a$ ,  $\mathbf{x}_b$ ,  $\mathbf{x}_c$  et  $\mathbf{x}_d$  sont les points de référence et  $\mathbf{x}_i$  un point inconnu à déterminer en fonction de ces quatre points de référence, l'expression de la jacobienne devient :

$$\mathbf{J}(\mathbf{x}_1, \dots, \mathbf{x}_n) = \begin{pmatrix} \mathbf{J}_1 & 0 & \dots & 0 \\ 0 & \mathbf{J}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{J}_n \end{pmatrix}, \quad (3.8)$$

avec les matrices diagonales ( $4 \times 3$ )

$$\mathbf{J}_i(\mathbf{x}_i, \mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, \mathbf{x}_d) = 4 \begin{pmatrix} f_{ia}(\mathbf{x}_i, \mathbf{x}_a)(\mathbf{x}_i - \mathbf{x}_a)^T \\ f_{ib}(\mathbf{x}_i, \mathbf{x}_b)(\mathbf{x}_i - \mathbf{x}_b)^T \\ f_{ic}(\mathbf{x}_i, \mathbf{x}_c)(\mathbf{x}_i - \mathbf{x}_c)^T \\ f_{id}(\mathbf{x}_i, \mathbf{x}_d)(\mathbf{x}_i - \mathbf{x}_d)^T \end{pmatrix}. \quad (3.9)$$

Cette optimisation du code nous permet d'obtenir des résultats significatifs : pour 31 points à déterminer (soit 93 inconnues) associés à 124 ( $4 \times 31$ ) contraintes souples et 30 contraintes dures, nous obtenons une fréquence d'affichage de 110 images par seconde dans le cas optimisé pour 51 images par seconde dans le cas non optimisé.

### 3.6.4 Application à un graphe cycle

La dernière expérience propose d'étendre la résolution à des problèmes plus larges tels que des graphes contenant des cycles. Pour cette catégorie d'applications nous avons utilisé un graphe constitué de vingt sommets, soit vingt distances dures correspondant aux segments entre les sommets adjacents. Nous avons également ajouté deux distances dures représentant le contact entre deux cibles et deux sommets de notre graphe. Enfin, nous avons mis en opposition deux expériences avec et sans contraintes souples afin de visualiser l'importance de celles-ci dans la déformation spatiale de notre graphe. Comme illustré par la figure 3.10, la méthode fonctionne bien pour des graphes cycles et nous permet d'opposer les résultats avec et sans contraintes souples. Cette démonstration met en avant le fait qu'à travers la manière de définir les contraintes souples, en plus de garantir des états de référence au cours de la manipulation, il nous est possible de donner un style à la déformation de la chaîne. Notamment il est possible de définir un style relatif à la rigidité de la structure considérée.

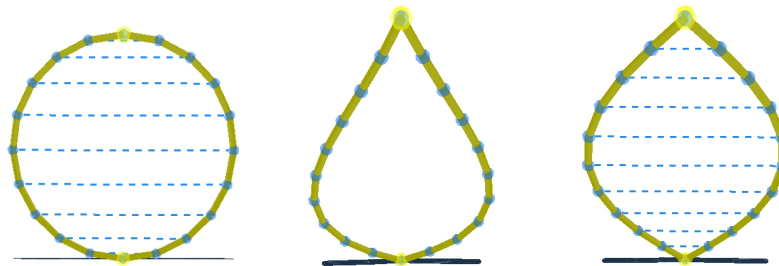


FIGURE 3.10 – Application à un graphe cycle. Sans contrainte souple (figure de gauche) et avec contraintes souples (figure de droite) : ces contraintes sont représentées par les pointillés en bleu

### 3.7 Discussion et perspectives

Les résultats obtenus démontrent la capacité de la méthode basée distances à résoudre des problèmes de contrôle cinématique du mouvement, dans des tâches diversifiées allant du simple suivi de mouvements capturés à la résolution de problèmes de cinématique inverse et à la modification de mouvements. L'originalité de la méthode réside essentiellement dans la possibilité de définir des tâches sous la forme d'un ensemble de contraintes géométriques qui traduisent de manière simple et pertinente la fonction à réaliser (par exemple adapter la main à l'objet à saisir). Ces contraintes s'expriment en particulier sous la forme de distances qui caractérisent la proximité de points du squelette (distances inter-articulations), ou d'effecteurs à effecteurs ou encore à d'autres points de l'environnement. La formulation du contrôle du mouvement que nous proposons permet ainsi de prendre en compte de manière implicite un ensemble de contraintes qui incluent notamment les contraintes liées à la tâche. L'incorporation de ces contraintes dans les méthodes d'optimisation fournit des solutions optimales et régulières, relativement robustes aux variations de conditions initiales et de tâches. La méthode générale est étendue, afin de séparer la contribution des distances dures (tâche principale) et des distances souples (tâche secondaire), permettant ainsi d'introduire une forme de priorité entre contraintes. De plus, la projection des solutions de la tâche secondaire sur le noyau de la tâche principale permet de garantir le respect de la tâche principale tout en réduisant l'espace des solutions. La méthode permet de résoudre toute une gamme de problèmes, incluant la gestion d'effecteurs multiples et de graphes cycles.

La méthode s'adapte facilement à des tâches d'adaptation du mouvement à partir de données capturées. Cependant, on observe des limitations de la méthode lorsque les contraintes spécifiées sont trop différentes de celles du mouvement original, ce qui peut conduire à une non convergence de l'algorithme, voire à des instabilités. Par ailleurs, nous avons vu que l'information caractérisant le squelette est redondante et sur-contraint le problème. Nous proposons dans la seconde partie de cette thèse (chapitre 6) une manière de résoudre ce problème en relâchant les contraintes souples. Cette opération s'effectue par une formulation linéaire des contraintes souples en utilisant l'opérateur Laplacien. La méthode présentée ne traite pas non plus les aspects temporels des animations. Des améliorations significatives seraient envisageables en couplant aux fonctions spatiales des fonctions temporelles modélisant les synergies articulaires [10]. Nous présenterons dans le chapitre 7 une méthode pour préserver lors de l'édition du mouvement les propriétés spatio-temporelles. Enfin, la méthode pourrait également être étendue afin de traiter des nuages de points issus de maillages ou de données capturées.

# 4

## Application à la recherche de mouvement

### Sommaire

---

<b>4.1 Introduction</b> . . . . .	<b>47</b>
<b>4.2 Travaux connexes</b> . . . . .	<b>48</b>
<b>4.3 Représentation du mouvement</b> . . . . .	<b>49</b>
<b>4.4 Méthodologie</b> . . . . .	<b>50</b>
<b>4.5 Résultats</b> . . . . .	<b>50</b>
<b>4.6 Conclusion</b> . . . . .	<b>52</b>

---

### 4.1 Introduction

Nous avons présenté une première méthode d'édition de mouvement à partir de sa représentation dans l'espace métrique. Nous avons vu en particulier que cette représentation amène à une alternative intéressante de la cinématique inverse classique et permet un meilleur contrôle de l'animation du squelette. Dans ce chapitre nous mettons en évidence un autre intérêt de caractériser le mouvement dans l'espace métrique et présentons une illustration de ce mode de représentation appliqué à la recherche de mouvement.

L'industrie de l'animation graphique fournit aujourd'hui un grand nombre de données issues de la capture de mouvement. Une problématique émergente consiste à trouver des méthodes permettant d'indexer et de retrouver des mouvements dans de grandes bases de données. L'objectif est le suivant : à partir d'un mouvement de référence, trouver l'ensemble des mouvements qui lui sont similaires. Dans ce chapitre nous présentons une nouvelle technique dédiée à la recherche de mouvements dans des collections de mouvements. Pour ce faire, il est classique d'associer une **représentation** du mouvement (par exemple dans l'espace des rotations ou des positions) avec une **mesure de similarité**.

Nous avons précédemment présenté l'utilisation de l'espace métrique pour de l'animation de squelette en appliquant en particulier le problème classique de géométrie des distances à la reconstruction de squelette. A travers cette première approche, nous avons proposé une

nouvelle représentation du squelette caractérisée par l'ensemble des distances entre ses articulations. Dans cette section, nous appliquons cette nouvelle représentation au contexte de la recherche de mouvement.

Dans un premier temps, nous mettons en évidence l'intérêt d'utiliser une représentation du mouvement par des distances entre les articulations des squelettes, couplée au produit scalaire dans l'espace latent. Nous verrons que cette approche permet une bonne caractérisation du mouvement et un coût de calcul particulièrement faible. Nous testerons enfin notre méthode sur la base de données HDM05 [128] qui est appropriée à notre approche, car elle contient un grand nombre de classes de mouvements étiquetés, chaque classe étant représentée par un nombre significatif d'exemples.

## 4.2 Travaux connexes

Les techniques de recherche de mouvement couplent en général une représentation du mouvement associée à une mesure de similarité. Une bonne représentation donne une bonne caractérisation du mouvement et améliore l'efficacité de la mesure de similarité. Cette efficacité se mesure en taux de réussite de la recherche mais également en coût de calcul. Ces mesures de similarité sont de deux natures : numérique ou logique. Parmi les approches numériques, les techniques de réduction de dimension ont l'avantage d'offrir une représentation plus petite que le mouvement initial sans trop de perte d'information. Alexa et Muller [4] sont les premiers à introduire la technique d'analyse par composantes principales (ACP), qui consiste à projeter un maximum d'informations sur l'axe qui décrit le mieux le mouvement (maximum de variance). En 2007, Xiang and Zhu [195] présentent une méthode non-linéaire utilisant une mise en correspondance de formes isométriques basée sur les distances géodésiques. Kovar et ses co-auteurs [99] proposent une méthode basée sur les distances entre les nuages de points décrivant les postures des mouvements. Ils utilisent l'algorithme de déformation temporelle dynamique (DTW) pour prendre en considération dans un même algorithme les relations temporelles et spatiales du mouvement. La DTW est un algorithme efficace, mais coûteux en temps de calcul : chaque mouvement est représenté par une matrice de dimension  $n \times n$  ( $n$  le nombre de postures dans le mouvement). D'autres approches comme la représentation géométrique associée à des lois logiques fournissent une caractérisation abstraite du mouvement. Müller et al. [130] effectuent ainsi un pré-traitement de la base de donnée en segmentant les mouvements en fonction de leurs caractéristiques géométriques. Plus tard Müller et al. [129] introduisent une représentation du mouvement définie par des matrices dont les lignes représentent les informations spatiales et les colonnes les informations temporelles. Plus récemment Liang et al. [107] ont présenté une méthode originale basée sur des contraintes de cinématique et de dynamique pour rechercher le mouvement.

### 4.3 Représentation du mouvement

Nous proposons d'utiliser les relations spatiales entre les positions des articulations pour caractériser le mouvement. Plus précisément, un mouvement est exprimé comme l'évolution des distances euclidiennes entre les articulations du squelette au cours du temps. Cet espace est désigné comme l'espace métrique.

**Représentation d'une posture** Nous reprenons ici la description faite dans la section 2.2 où chaque posture est représentée par le graphe  $G = (V, E, d)$ .  $V$  est l'ensemble des sommets et  $E$  est l'ensemble des arêtes évoluant dans le temps. Pour cette étude, les segments qui composent le squelette étant de longueurs fixes lors de l'animation, cette information est redondante et ne caractérise donc pas le mouvement. Contrairement aux chapitres précédents, nous ne l'incluons pas dans  $E$ . Une illustration de ce graphe est donnée par la figure 4.1.

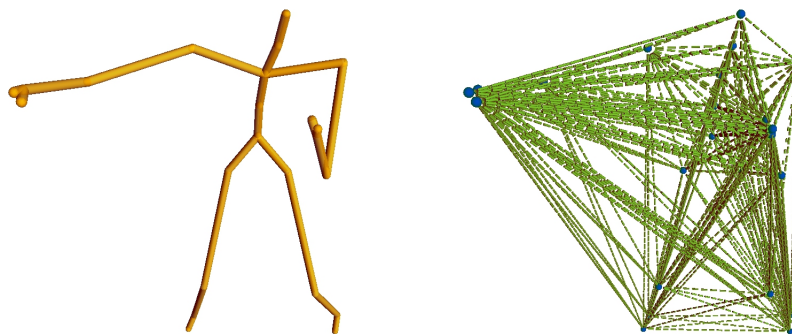


FIGURE 4.1 – Création du graphe associé à une posture : en jaune le squelette (ses segments ne sont pas inclus au graphe  $G$ ) et en pointillé vert les arêtes de longueurs variables.

**Représentation du mouvement** Le mouvement est représenté par une matrice  $M$  encodant les informations de distances des graphes de chaque posture du mouvement.  $M$  est donc de dimension  $n \times m$  où  $n$  est le nombre de distances et  $m$  le nombre de postures composant le mouvement. Contrairement aux représentations par les positions ou par les rotations, il est intéressant d'observer qu'une quantité plus importante de données est nécessaire pour décrire une posture. Par exemple, dans le cas d'un squelette avec 31 articulations, 465 valeurs de distances sont nécessaires pour définir la posture, seulement  $31 * 3 = 93$  valeurs sont exploitées dans l'espace des positions.

L'information caractérisant un mouvement étant la plupart du temps redondante (orientations des articulations, positions ou distances), il est courant d'utiliser des méthodes statistiques pour réduire l'espace de représentation. En particulier, les méthodes classiques utilisent l'analyse en composantes principales (ACP) pour décomposer le mouvement en un sous-espace orthogonal, dans lequel chacune des composantes principales décrit une quantité décroissante



de la variance. Plus précisément, nous considérons la matrice de covariance  $\tilde{M}$  de  $M$  de dimension  $n \times n$  et sa décomposition en valeurs singulières (SVD) telle que

$$\tilde{M} = U\Sigma U^T. \quad (4.1)$$

Cette approche permet de compresser les informations du mouvement, et de donner une représentation simplifiée et significative du mouvement à travers les premiers axes propres. **Nous faisons ici l'hypothèse que l'espace de grande dimension induit par les distances s'adapte particulièrement bien à ces techniques de réduction de dimension. De plus nous pensons que la forte corrélation entre les données permet une décomposition du mouvement fortement répartie sur les premiers axes de l'ACP.** Cette hypothèse est inspirée par la théorie des méthodes à noyau classique qui opèrent un changement d'espace (induit par une fonction noyau, le *kernel*) avant d'appliquer des méthodes d'analyse statistique des données (*Kernel* ACP par exemple [146]).

Notons également que cette représentation est **invariante aux translations et rotations**. Cela permet en particulier d'éviter d'avoir à aligner les données au préalable.

## 4.4 Méthodologie

Afin de trouver les mouvements qui sont similaires au mouvement original dans la base de données, on utilise une mesure de similarité qui est basée sur le premier axe de la méthode ACP appliquée à la matrice de distances caractérisant chaque mouvement. Plus précisément, pour deux mouvements représentés par les matrices  $M_i$  et  $M_j$ , la mesure de similarité est définie par le produit scalaire  $s(\mathbf{V}_i, \mathbf{V}_j)$  entre les principaux vecteurs  $\mathbf{V}_i$  et  $\mathbf{V}_j$  :

$$s(\mathbf{V}_i, \mathbf{V}_j) = \mathbf{V}_i \cdot \mathbf{V}_j \quad (4.2)$$

avec  $\mathbf{V}_i$  le premier vecteur singulier extrait par une décomposition en valeurs singulières (SVD) appliqué à  $\tilde{M}_i$ . Ainsi si  $\tilde{M}_i = U_i \Sigma_i U_i^T$  alors  $\mathbf{V}_i$  est la première colonne de  $U_i$ . **Notons ici que cette mesure permet de travailler indépendamment de la longueur temporelle des séquences considérées.**  $\mathbf{V}_i$  est de dimension  $n$  qui est égal au nombre de distances caractérisant une posture. Cette caractéristique est commune à l'ensemble des mouvements ( $\dim(\mathbf{V}_i) = \dim(\mathbf{V}_j) = n, \forall (i, j)$ ).

## 4.5 Résultats

Notre méthode a été testée sur la base de données de mouvement *HDM05* [128], qui contient une collection de 497 mouvements répartis dans 78 classes différentes. La figure 4.2 illustre l'exemple de 3 cas de recherche associé à leurs résultats.

Pour tester la fiabilité de notre méthode, notre démarche est de retirer un à un les mouve-

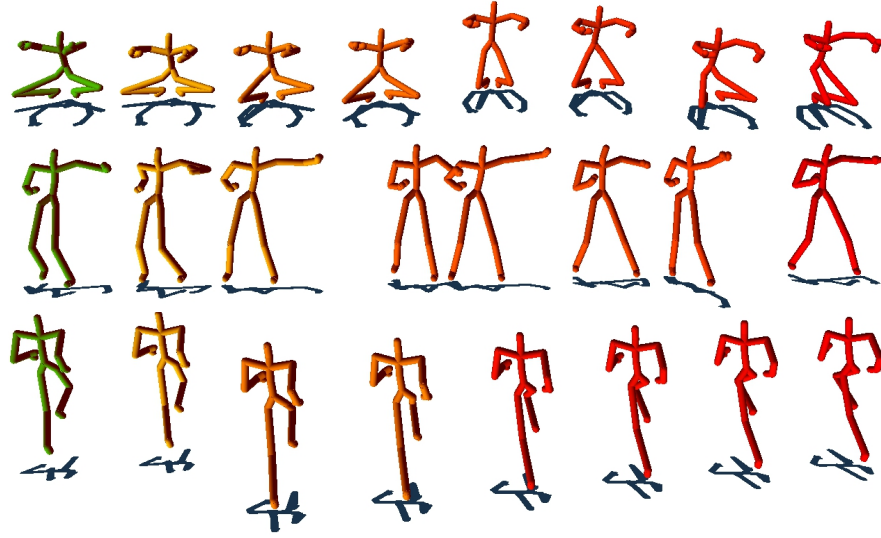


FIGURE 4.2 – Résultats obtenus pour les mouvements *HDM\_bd\_squat1Reps*, *HDM\_bd\_punchLSide1Reps* et *HDM\_bd\_hopLLeg2hops* : les 7 premiers candidats trouvés. En vert le mouvement de référence, et de jaune (meilleurs résultat) à rouge (plus mauvais candidat) les candidats dans l'ordre de succès.

ments de la base de données, puis de les comparer au reste de la base. Pour chaque mouvement, nous appliquons la méthode des plus proches voisins en testant son appartenance à une catégorie. Ainsi, nous construisons une matrice de confusion pour évaluer les résultats. Une entrée  $c_{i,j}$  de la matrice de confusion correspond au nombre de fois qu'un mouvement de classe  $i$  est associé à un mouvement de classe  $j$  normalisé par le nombre de mouvements dans la classe  $i$  :

$$c_{i,j} = \frac{\sum_{k \in \mathcal{N}(i)} f_j(k)}{|\mathcal{N}(i)|}, \quad (4.3)$$

où  $\mathcal{N}(i)$  est le nombre de mouvements dans la classe  $i$ .  $f_j$  est égale à 1 si le mouvement  $k$  est un succès avec la méthode du plus proche voisin avec la classe  $j$ , 0 sinon. Le résultat souhaité est donc d'avoir une répartition des valeurs élevée le long de la diagonale (soit proche de 1). Nous avons effectué le test pour tous les mouvements sans aucun traitement préalable. Les résultats illustrés par la figure 4.3 montrent les matrices de confusion lors de la recherche d'un mouvement représenté par le premier vecteur propre résultant de l'ACP, appliqué aux distances (a), aux positions (b), et aux orientations (c). Plus précisément, la recherche est effectuée pour les deux premiers voisins les plus proches. Ce choix est justifié par le nombre de mouvement contenu dans chaque classe qui est au minimum de 3. Nous observons une forte cohérence entre les mouvements recherchés et les résultats obtenus dans le cas des mouvements représentés par des distances. A l'inverse il existe peu de corrélation pour les recherches effectuées avec des mouvements représentés par des positions ou des rotations. Ces résultats montrent

l'intérêt d'utiliser des distances pour représenter un mouvement par rapport aux positions ou aux rotations. Le taux d'efficacité de notre algorithme est en moyenne de 68,8% de succès sans opération de transformation préalable sur la base de données (13,7 % pour les positions et 13,9 % pour les rotations). Par ailleurs, la base de données HDM05 proposant des mouvements de toutes natures, nous pouvons observer à travers les matrices de confusion (figure 4.3) que notre méthode s'applique à tout type de mouvement. Nous avons toutefois remarqué que les résultats sont logiquement meilleurs pour des mouvements bien segmentés. A l'inverse les longues séquences enchaînant plusieurs mouvements donnent des taux moins bons.

Enfin il est particulièrement intéressant de remarquer qu'un simple produit scalaire entre les deux premiers vecteurs propres (de dimension 465 ici) permet des temps de calcul négligeables.

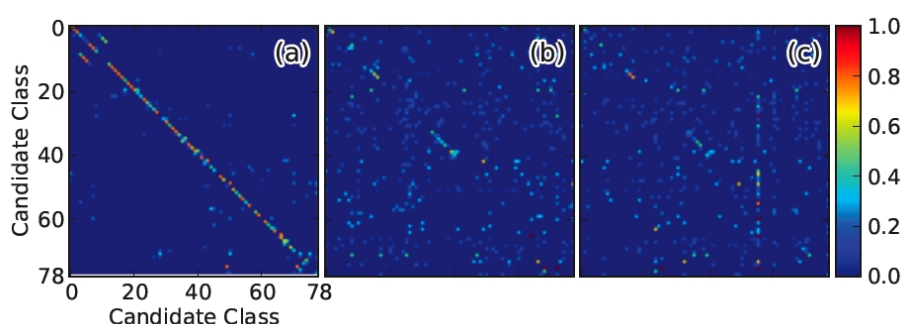


FIGURE 4.3 – Les matrices de confusion obtenues en utilisant les représentations des mouvements par des distances (a), positions (b) et rotations (c)

## 4.6 Conclusion

Nous avons présenté dans ce chapitre une méthode de recherche de mouvement simple et efficace dédiée à la recherche de mouvements similaires à un mouvement exemple dans une grande base de données. Notre méthode est basée sur une représentation du mouvement dans l'espace métrique, un mouvement étant caractérisé par l'évolution au cours du temps des distances entre les articulations du squelette. Le mouvement est alors réduit en utilisant une analyse en composantes principales. Une telle approche permet une recherche du mouvement dans la base de données simplement et rapidement par le produit scalaire des premiers axes ACP de chaque paire de mouvement. Nous avons montré qu'il est intéressant de représenter un mouvement dans l'espace des distances, en particulier parce que les techniques de réduction de dimension nécessitent une forte corrélation entre les données. Cette propriété est directement liée à la grande quantité d'information caractérisant le mouvement lorsqu'on s'intéresse à une représentation basée distances (en opposition avec les positions et les rotations). Par ailleurs notre méthode permet d'éviter d'avoir à aligner les données des mouvements au préalable puisque la représentation d'une posture dans l'espace des distances est invariante aux

translations et rotations. Enfin, notre technique qui est temps réel est efficace sur des mouvements sans pré-traitement, puisqu'elle offre un taux de réussite de près de 70% sur tout type de mouvement.

Bien qu'intéressante, cette méthode est davantage une illustration pertinente de l'utilisation des distances pour la représentation du mouvement. Par ailleurs, nous pensons qu'il serait intéressant de compléter cette étude par des comparaisons avec les méthodes présentées dans l'état de l'art, notamment en terme de taux de réussite et temps de recherche.

Parmi les perspectives, nous pensons qu'il est possible d'améliorer les résultats en définissant un produit scalaire à partir de plusieurs vecteurs propres. Cette technique nécessiterait toutefois un alignement des deux bases à comparer opérable par une rotation (une telle technique est proposée par [167]). Une autre possibilité consisterait à comparer l'ensemble des vecteurs propres entre eux en utilisant le produit scalaire de Frobenius appliqué à l'alignement de noyau [44]. Nous pensons par ailleurs qu'un pré-traitement de la base de données destiné à segmenter les mouvements améliorerait significativement les résultats. Enfin dans un tout autre contexte, nous pensons que la représentation que nous avons introduite pourrait s'appliquer au domaine de la recherche en temps réel de maillage (entiers ou partiels). En effet, la recherche de maillage pose le problème d'identifier des maillages visuellement similaires mais dont la géométrie et/ou la topologie sont différentes (ces maillages ayant éventuellement subi des transformations de type rotation ou translation). L'intérêt de notre représentation est qu'elle répond à ces problématiques puisque nous ne prenons pas en considération les aspects topologiques, ainsi qu'une éventuelle transformation rigide appliquée au graphe.



## **Deuxième partie**

# **Représentation Laplacienne du mouvement associée aux distances comme contrainte implicite**



# 5

## L'opérateur Laplacien pour l'animation 3D

### Sommaire

---

<b>5.1 Introduction</b> . . . . .	<b>57</b>
<b>5.2 Présentation de l'opérateur Laplacien</b> . . . . .	<b>58</b>
5.2.1 L'opérateur Laplacien continu . . . . .	58
5.2.2 L'opérateur Laplacien discret : application à l'édition de graphe . . . . .	59
<b>5.3 Applications à l'informatique graphique</b> . . . . .	<b>66</b>
5.3.1 Traitement de maillage . . . . .	66
5.3.2 Édition de mouvement . . . . .	72
<b>5.4 Conclusion</b> . . . . .	<b>75</b>

---

### 5.1 Introduction

Utiliser la notion de relation spatiale entre les différents éléments d'une même scène pour animer des squelettes est une alternative intéressante aux techniques classiques. Parmi les avantages d'un tel concept, nous retenons que notre approche nous a permis de mieux préserver les notions de proximité entre les articulations d'un squelette durant l'édition, comme par exemple la préservation des contacts d'une main avec l'environnement ou entre plusieurs articulations d'un même squelette. Un tel concept permet également de contrôler le mouvement d'un plus haut niveau que les techniques classiques comme la cinématique inverse.

Toutefois, nous avons vu que l'utilisation de l'espace des distances comme moyen unique d'édition et de résolution du mouvement est limitée. En effet, représenter l'intégralité d'une posture uniquement avec les distances sur-contraint fortement la méthode de résolution, ce qui influe directement sur la stabilité de la solution puisque l'espace des distances est non linéaire. Par ailleurs, un tel concept, à l'instar de la cinématique inverse, traite les postures indépendamment les unes des autres, ce qui est un inconvénient pour la préservation de la dynamique du mouvement lors de l'édition.

Dans cette partie du manuscrit, nous introduisons un nouvel outil qui a également pour objectif de préserver les relations spatiales entre les différents éléments d'une même scène :



*l'opérateur Laplacien*. Plus précisément l'opérateur Laplacien permet à travers une information différentielle d'exprimer un objet vis-à-vis de son voisinage. Nous verrons toutefois que cette nouvelle étude est dans le prolongement de la précédente puisque la notion de distance est indispensable, notamment pour préserver les longueurs des segments du squelette, ou bien pour appliquer des contraintes géométriques exprimées sous forme de distances pour des tâches spécifiques.

Dans cette section, nous présentons dans un premier temps l'opérateur Laplacien continu avant d'introduire sa version discrétisée. Puis nous présenterons les différentes utilisations de l'opérateur Laplacien en informatique graphique, et nous discuterons de son application dans le cadre de l'animation de squelette et de maillage.

## 5.2 Présentation de l'opérateur Laplacien

L'opérateur Laplacien étant fondamental dans les lois de la physique, on le trouve dans tout processus de diffusion et de propagation d'ondes et de phénomènes vibratoires. Ainsi cet opérateur apparaît dans la formulation mathématique de nombreuses disciplines comme la géophysique, l'électrostatique, la thermodynamique, la mécanique classique et quantique. Comme nous le verrons dans la section suivante, il est également très utilisé en informatique graphique, notamment pour l'édition et l'analyse de formes géométriques.

Dans cette section, nous présentons mathématiquement l'opérateur Laplacien continu et discret à travers plusieurs définitions et exemples.

### 5.2.1 L'opérateur Laplacien continu

**L'opérateur gradient  $\nabla$  :** l'opérateur Laplacien utilise l'opérateur gradient  $\nabla$  (prononcé "nabla") qui peut être exprimé par le vecteur des dérivées partielles :

$$\nabla := \left( \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_n} \right). \quad (5.1)$$

Appliquer  $\nabla$  à une fonction scalaire  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  donne le vecteur *gradient* :

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right), \quad (5.2)$$

et en appliquant le produit scalaire de  $\nabla$  avec une fonction  $f$ , on obtient la *divergence* :

$$\nabla \cdot f = \sum_i^n \frac{\partial f}{\partial x_i}. \quad (5.3)$$

**L'opérateur Laplacien  $\Delta$  :** ainsi, dans sa forme continue en coordonnées cartésiennes dans  $\mathbb{R}^n$ , l'opérateur Laplacien appliqué à une fonction  $f$  deux fois dérivable est la divergence du

gradient de  $f$  et s'écrit :

$$\Delta f = \operatorname{div} \nabla f = \sum_i^n \frac{\partial^2 f}{\partial x_i^2}, \quad (5.4)$$

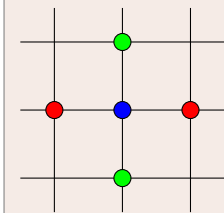
où  $\Delta$  est l'opérateur Laplacien (ou encore  $\Delta = \nabla \cdot \nabla = \nabla^2$ ). Par exemple en coordonnées cartésiennes tridimensionnelles, le Laplacien est :

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}. \quad (5.5)$$

## 5.2.2 L'opérateur Laplacien discret : application à l'édition de graphe

### 5.2.2.1 Exemples de cas discrets

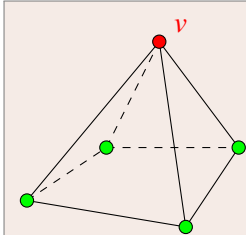
L'opérateur Laplacien discret défini sur un graphe ou une grille discrétisée est analogue à sa version continue. Les exemples présentés dans les figures 5.1 et 5.2 qui sont respectivement appliqués à une grille 2D et un graphe quelconque 3D montrent l'expression de l'opérateur Laplacien dans le cas discret.



$$\Delta f(x_i, y_i) = \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{h^2} + \frac{f(x_{i-1}, y_i) - f(x_i, y_i)}{h^2} + \frac{f(x_i, y_{i+1}) - f(x_i, y_i)}{h^2} + \frac{f(x_i, y_{i-1}) - f(x_i, y_i)}{h^2} \quad (5.6)$$

où  $h$  est de valeur égale à la distance entre deux mailles adjacentes.

FIGURE 5.1 – Discrétisation des différences finies pour une grille en 2 dimensions



$$\Delta f(v) = \sum_{v_i \in N_1(v)} (f(v) - f(v_i)) = |N_1(v)|f(v) - \sum_{v_i \in N_1(v)} f(v_i) \quad (5.7)$$

où  $N_1(v)$  est l'ensemble des sommets voisins de  $v$ , apparaissent en vert.

FIGURE 5.2 – Exemple de l'expression de l'opérateur Laplacien discret appliqué à un graphe 3D quelconque

### 5.2.2.2 Notations et définitions

Considérons une fonction réelle deux fois dérivable. Les formules de Taylor permettent d'obtenir une approximation du Laplacien  $\Delta f(x) = \frac{\partial^2 f(x)}{\partial x^2}$  sur  $\mathbb{R}$  appelée schéma des différences

finies. Pour  $h$  petit, on a

$$\Delta f(x) \approx \frac{2}{h^2} \left( \frac{f(x+h)+f(x-h)}{2} - f(x) \right), \quad (5.8)$$

et cette relation se généralise aux dimensions supérieures. On remarque qu'il s'agit d'une différence entre la moyenne des valeurs prises par la fonction sur les points voisins et la valeur de la fonction sur le point courant. On va utiliser cette idée pour étendre la définition aux graphes.

**Application au cas d'un graphe** Nous pouvons généraliser les définitions vues précédemment pour les appliquer au cas discret sur des graphes quelconques. Des études détaillées sur ce point ont été menées par Sorkine [156, 157]. Soit le graphe  $G = (\mathbf{V}, \mathbf{E})$  où  $\mathbf{V}$  est l'ensemble des sommets et  $\mathbf{E}$  décrit la connectivité du graphe (ensemble d'arêtes de  $G$ ). Notons également que l'ensemble des voisins adjacents d'un sommet d'indice  $i$  est donné par la terminologie suivante :  $\mathcal{N}(i) = \{v_j | (v_i, v_j) \in E\}$  et le degré  $d_i$  d'un sommet est déterminé par le nombre d'éléments dans  $\mathcal{N}(i)$ , ( $d_i = |\mathcal{N}(i)|$ ). Pour une fonction  $f : \mathbf{V} \rightarrow \mathbb{R}$ , le Laplacien de  $f$  est défini dans le cas général par :

$$\Delta f(\mathbf{v}_i) = \sum_{j \in \mathcal{N}(i), d(v_i, v_j)=1} (f(\mathbf{v}_i) - f(\mathbf{v}_j)). \quad (5.9)$$

où  $d(\mathbf{v}_i, \mathbf{v}_j)$  est la distance entre  $\mathbf{v}_i$  et  $\mathbf{v}_j$ . Dans notre cas  $\mathbf{V} = [\mathbf{v}_1^T, \mathbf{v}_2^T, \dots, \mathbf{v}_n^T]^T$ ,  $\mathbf{v}_i \in \mathbb{R}^3$  où  $n$  est égal au nombre de sommets. Nous considérons alors que  $f(\mathbf{v}_i) = \mathbf{v}_i$  et travaillons avec des graphes pondérés (un poids est associé à chaque arête). Nous cherchons donc à exprimer le graphe par une information différentielle, soit un ensemble de coordonnées différentielles  $\{\delta_i\}$ .  $\delta_i$  ( $\delta_i \in \mathbb{R}^3$ ) est le Laplacien de  $\mathbf{v}_i$  et plus précisément la coordonnée  $i$  est représentée par la différence entre  $\mathbf{v}_i$  et la moyenne de ses voisins :

$$\delta_i = \mathcal{L}(\mathbf{v}_i) = \sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{v}_i - \mathbf{v}_j), \quad (5.10)$$

où  $\mathcal{L}$  est l'opérateur Laplacien discret appliqué à  $\mathbf{v}_i$ .  $w_{ij}$  peut prendre plusieurs formes selon l'importance que l'on veut donner à un sommet voisin vis à vis des autres. Ces poids sont également dépendants des propriétés du graphe. Par exemple :

- **les poids uniformes** donnent la même importance à l'ensemble des voisins du sommet  $i$  :

$$w_{ij} = 1, \quad (5.11)$$

- **les poids cotangents** :

$$w_{ij} = \frac{1}{2} (\cot \alpha_{ij} + \cot \beta_{ij}), \quad (5.12)$$

sont utilisés pour la construction de l'opérateur Laplace-Beltrami. Ils sont applicables sur des graphes particuliers tels que ceux représentant des surfaces triangulées 2-variété. Ainsi toutes les arêtes sont communes à exactement deux faces (le maillage est sans trou et orientable). Une illustration de leur détermination est donnée par la figure 5.3.

– **Les poids suivant un noyau de diffusion Gaussien :**

$$w_{ij} = \exp\left(-\frac{|\mathbf{v}_i - \mathbf{v}_j|^2}{\gamma}\right) \quad (5.13)$$

donnent une bonne estimation de l'opérateur Laplace-Beltrami [40] quand  $\gamma \rightarrow 0$ . Le paramètre  $\gamma$  peut être automatiquement déterminé comme la distance médiane des plus proches voisins.

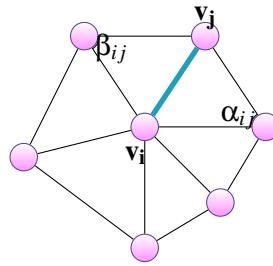


FIGURE 5.3 – Les angles utilisés dans le calcul des poids cotangents pour l'arête  $(i, j)$  (en bleu).

On peut également trouver ces poids sous leur forme normalisée où :

$$w_{ij} = \frac{w_{ij}}{d_i} \quad \text{ou} \quad (5.14)$$

$$w_{ij} = \frac{w_{ij}}{\sum_{k \in \mathcal{N}(i)} w_{ik}}, \quad (5.15)$$

qui sont respectivement applicables au poids uniformes dans le premier cas et aux poids cotangents et suivant un noyau de diffusion dans le second cas.

**Forme matricielle** La transformation entre  $\mathbf{V}$ , vecteur des coordonnées cartésiennes en absolu et  $\Delta$  peut être mise sous forme matricielle. Soit  $\mathbf{A}$ , la matrice de l'adjacence (connectivité) de  $G$  :

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{si } (\mathbf{v}_i, \mathbf{v}_j) \in E \\ 0 & \text{sinon} \end{cases}$$

et  $D$  la matrice diagonale tel que  $D = \text{diag}(d_1, \dots, d_n)$  est la matrice des degrés. Ainsi la matrice qui transforme les coordonnées absolues en coordonnées relatives s'exprime par :

$$L = I - D^{-1}A, \quad (5.16)$$

Finalement l'équation (5.10) peut être mise sous la forme

$$\Delta = LV. \quad (5.17)$$

La matrice  $L$  est considérée comme l'opérateur Laplacien du graphe  $G$  avec la connectivité  $A$ . La matrice Laplacienne de dimension  $n \times n$  ( $n$  étant égal au nombre de sommets) pour l'intégralité d'un graphe se formule également :

$$L_{ij} = \begin{cases} \sum_{v_j \in \mathcal{N}(i)} w_{ij} & i = j \\ -w_{ij} & (v_i, v_j) \in E \\ 0 & \text{sinon} \end{cases}$$

La matrice Laplacienne est aussi appelée matrice topologique du graphe [55]. Nous remarquons par ailleurs que  $L$  est symétrique et définie semi positive.

La figure 5.4 illustre la visualisation des coordonnées Laplaciennes pour  $L$  avec des poids uniformes de différents maillages. Il est intéressant de remarquer que dans certains cas ces coordonnées se rapprochent des normales au maillage. On peut ici établir le lien avec le Laplacien à poids cotangents qui propose une bonne approximation de la normale à la surface d'un maillage.

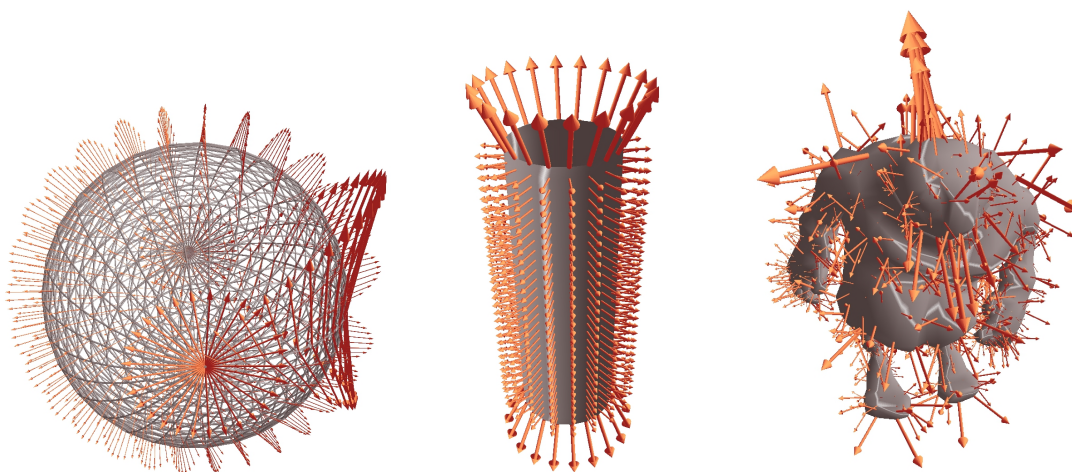


FIGURE 5.4 – Coordonnées différentielles associées à i) une sphère associée à une mauvaise connectivité ii) un cylindre iii) au modèle *Biggy*

### 5.2.2.3 Reconstruction du graphe à partir de $\Delta$

Les coordonnées Laplaciennes sont invariantes aux translations, mais sensibles aux transformations linéaires.  $L$  est de rang  $n - 1$ , ce qui signifie que  $V$  peut être retrouvé à partir de  $\Delta$  en fixant un point et en résolvant le système linéaire. D'un autre côté, fixer plusieurs points revient à sur-contraindre le système, ce qui amène à une résolution donnant lieu à une solution approchée.

A partir d'un graphe dont l'ensemble des sommets sont connectés, l'idée pour éditer le graphe en utilisant les coordonnées laplaciennes  $\Delta$  revient à fixer par substitution plusieurs sommets à travers la formulation :

$$v'_i = u_i, i \in \{m, \dots, n\}, m < n \quad (5.18)$$

et en résolvant pour le reste des sommets  $v'_i, i \in \{1, \dots, m - 1\}$  en minimisant les coordonnées Laplaciennes de la géométrie  $V'$  avec les coordonnées laplaciennes  $\Delta$ . La fonction erreur  $E_L$  qui permet de trouver le vecteur  $V'$  peut être exprimée par :

$$E(V') = \sum_{i=1}^n \|\delta_i - \mathcal{L}(v'_i)\|^2 + \sum_{i=m}^n \|v'_i - u_i\|^2. \quad (5.19)$$

Il est important de noter que nous utilisons les contraintes de position (définies par l'équation (5.18)) au sens des moindres carrés : au lieu de les substituer dans le système linéaire, nous les *ajoutons* comme de nouvelles équations au système. Cette formulation des contraintes permet une meilleure restitution de la géométrie mais ne respecte pas complètement les contraintes de position.

La figure 5.5 montre un exemple simple d'édition de graphe en utilisant l'opérateur Laplacien avec des poids uniformes. La figure de gauche représente le graphe dans sa configuration spatiale de référence. Les sommets en rouge sont fixes, celui en vert est celui que l'on désire éditer et ceux en gris sont ceux non contraints. A droite, après le déplacement du point de contrôle (en vert), nous observons bien la préservation de la relation spatiale entre chaque sommet et ses voisins durant l'édition.

**Forme matricielle** La précédente équation peut être mise sous la forme matricielle :

$$\begin{pmatrix} L \\ w_u I_U \end{pmatrix} V' = \begin{pmatrix} \Delta \\ w_u U \end{pmatrix} \quad (5.20)$$

où  $I_U$  est la matrice associée aux contraintes contenues dans le vecteur  $U$  (contraintes de position) et  $w_u > 0$  est utilisé pour paramétrer l'importance des contraintes de distance.

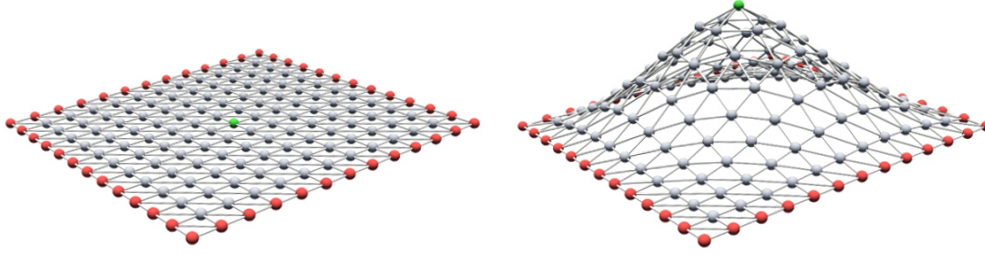


FIGURE 5.5 – Exemple d'édition de graphe appliqué à un maillage. À gauche le maillage initial, à droite le maillage déformé. En rouge les sommets fixes, en vert le sommet édité et en gris les sommets non contraints résultats de la minimisation.

#### 5.2.2.4 Résolution numérique

Le précédent système est de rang plein et a donc une unique solution au sens des moindres carrés. Le vecteur solution peut être calculé par une factorisation de Cholesky. En notant  $\mathbf{A}$  la matrice rectangulaire de dimension  $(n + m) \times n$  concaténant  $\mathbf{L}$  avec  $w_u \mathbf{I}_U$  et  $\mathbf{b}$  le vecteur concaténant  $\Delta$  avec  $w_u \mathbf{U}$  alors le système peut être ré-écrit :

$$(\mathbf{A}^T \mathbf{A}) \mathbf{V}' = \mathbf{A}^T \mathbf{b}. \quad (5.21)$$

La matrice  $\mathbf{M} = \mathbf{A}^T \mathbf{A}$  est carrée, symétrique et définie positive. La factorisation de Cholesky nous permet d'obtenir  $\mathbf{M} = \mathbf{A}^T \mathbf{A} = \mathbf{R}^T \mathbf{R}$  telle que  $\mathbf{R}$  est une matrice triangulaire inférieure et nous pouvons donc résoudre par "back-substitution" les systèmes :

$$\mathbf{R}^T \mathbf{y} = \mathbf{A}^T \mathbf{b} \quad (5.22)$$

$$\mathbf{R} \mathbf{x} = \mathbf{y}. \quad (5.23)$$

De plus la matrice  $\mathbf{A}$  étant fortement creuse, nous utilisons la librairie Cholmod spécifique à la résolution par Cholesky de matrices creuses pour résoudre le système.

#### 5.2.2.5 Propriétés spectrales

L'opérateur Laplacien discret d'un graphe et sa décomposition spectrale (valeurs et vecteurs propres) ont plusieurs propriétés importantes qui en font un opérateur couramment utilisé dans les méthodes spectrales. Nous allons discuter de certaines de ces propriétés dans cette section.

Nous considérons le graphe pondéré  $G = (\mathbf{V}, \mathbf{E}, w)$  connexe et fini précédemment introduit et  $n$  le nombre de sommets de  $G$ . L'ensemble des fonctions réelles définies sur  $\mathbf{V}$  est alors un espace vectoriel de dimension  $n$  que l'on identifie à  $\mathbb{R}^n$  muni du produit scalaire  $\langle \cdot, \cdot \rangle$ . La représentation matricielle  $\mathbf{L}$  de dimension  $n \times n$  est symétrique et définie semi positive. Le spectre de  $\mathbf{L}$  est donc composé de  $n$  valeurs propres réelles  $\{\lambda_i\}_{i=0, \dots, n-1}$  rangées par ordre croissant. Chaque valeur propre  $\lambda_i$  est associée à un vecteur propre  $\mathbf{e}_i$  tel que  $\mathbf{L}$  possède une

famille de vecteurs propres  $\{e_i\}_{i=0,\dots,n-1}$  orthonormée.

Nous pouvons énumérer un certain nombre de propriétés liées à la décomposition spectrale du Laplacien :

- un vecteur propre  $e_i$  de  $L$  vérifie l'égalité  $Le_i = \lambda_i e_i$ ;
- la première et plus petite valeur propre  $\lambda_0 = 0$  est associée à un vecteur constant  $\mathbf{1}_n$  dont l'ensemble des coordonnées sont égales à 1 ( $\mathbf{1}_n = (1, \dots, 1)^T$ ,  $L\mathbf{1}_n = 0$ );
- le premier vecteur propre  $e_0 = \mathbf{1}_n$  est le vecteur décrivant une fonction constante caractérisant  $G$ . Ce vecteur implique en particulier que l'ensemble des sommets du graphe possède au moins un voisin ;
- la première valeur propre non nulle appelée *trou spectral* est particulièrement petite et la plus grande est inférieure ou égale à 2 ; ainsi  $\{\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{n-1}\} \subset [0, 2]$ .
- nous remarquons que les premiers vecteurs propres (correspondant aux petites valeurs propres) correspondent aux fréquences faibles alors que les plus élevées correspondent aux fréquences hautes.

De plus amples informations sont données par les travaux de Spielman et Teng [161].

L'étude spectrale est couramment utilisée en informatique graphique pour décrire des graphes et des maillages. Par exemple à l'image de l'utilisation de la décomposition de Fourier en traitement d'image pour décrire une image, on peut reconstruire un maillage à partir de sa décomposition spectrale : figure 5.6. L'étude spectrale permet également de faire du partitionnement de graphe ou encore de la mise en correspondance de graphe. Enfin, l'étude spectrale a récemment été utilisée en animation pour de l'interpolation entre différentes configurations d'un même graphe (cf section 5.3.2.3).

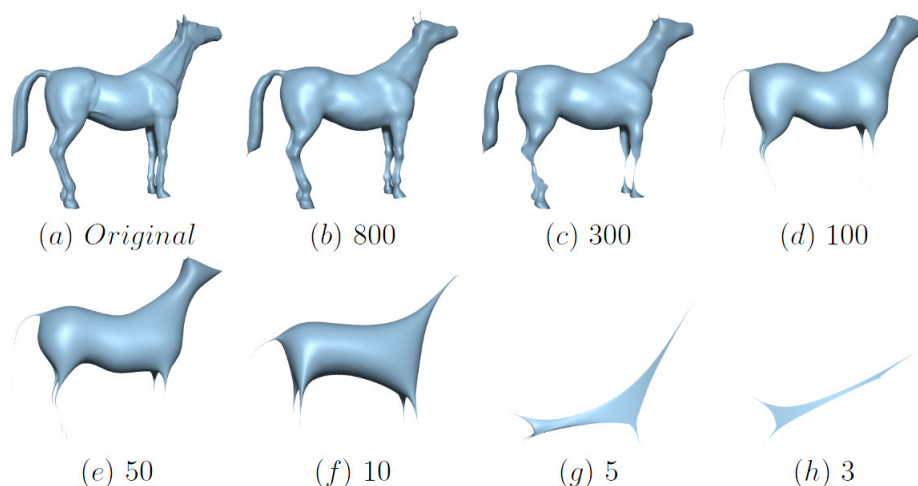


FIGURE 5.6 – Reconstruction du modèle *Horse* avec le nombre de vecteurs propres indiqués [48]



## 5.3 Applications à l'informatique graphique

Ces dernières années l'opérateur Laplacien a été utilisé dans plusieurs domaines de l'informatique graphique. Ses propriétés étant particulièrement adaptées aux traitements sur les surfaces, il a été employé pour la représentation, la paramétrisation, l'édition, le filtrage, la correspondance de formes, etc. et a été appliqué à des géométries 2D et 3D comme les images et les maillages. Nous détaillons ci-après les approches couramment employées dans le cadre du traitement de maillage et de l'édition de mouvement.

### 5.3.1 Traitement de maillage

De nombreuses études ont été menées pour de l'édition, de l'analyse et de la déformation de structures géométriques. Un maillage est classiquement représenté par un ensemble de positions absolues dans l'espace 3D. Il est intéressant de noter que cette représentation n'est pas idéale pour ce genre d'opération, une description locale de la géométrie indépendante du placement de l'objet dans l'espace Euclidien peut s'avérer plus intéressante. Dans cette section, nous revenons sur l'utilisation des caractéristiques de l'opérateur Laplacien discret dans le traitement de maillages. Le lecteur qui cherche des informations complémentaires sur les travaux existants peut se référer aux états de l'art de Grinspun [62] dédié à la géométrie différentielle et celui proposé par Sorkine [157] regroupant l'ensemble des applications liées à l'opérateur Laplacien sur le traitement de maillage.

**Principe** Rappelons que la coordonnée Laplacienne  $\delta_i$  d'un sommet  $v_i$  mesure la différence entre  $v_i$  et la moyenne de ses voisins qui lui sont adjacents :

$$\delta_i = \mathcal{L}(v_i) = v_i - \frac{1}{|\mathcal{N}(i)|} \sum_{v_j \in \mathcal{N}(i)} v_j. \quad (5.24)$$

Cette mesure définie plus en détail dans la section 5.2.2 est l'idée centrale des traitements appliqués sur les maillages. Notons qu'un maillage triangulaire de  $n$  sommets est défini par  $\mathcal{M} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$  où  $\mathbf{V}$  décrit l'ensemble des positions des sommets,  $\mathbf{E}$  l'ensemble des arêtes et  $\mathbf{F}$  l'ensemble des faces. Par ailleurs, le maillage  $\mathcal{M}$  étant un graphe particulier, nous pouvons lui appliquer l'ensemble des précédentes définitions vues dans la section 5.2.2.

#### 5.3.1.1 Édition de maillage

L'utilisation de l'opérateur Laplacien pour l'édition de maillage surfacique permet d'obtenir des déformations visuellement naturelles préservant les détails géométriques. L'édition de maillage surfacique (ensemble de triangles) ou volumique (ensemble de tétraèdres) se fait classiquement par une sélection de la zone à éditer (appelée zone d'intérêt : ROI) puis par une sélection des sommets fixes et ceux à déplacer comme contraintes de position. L'objectif de la

méthode de déformation est de préserver la forme d'origine du maillage tout en préservant les contraintes de position. Plusieurs propriétés caractérisent les méthodes de déformation :

- l'opérateur Laplacien est invariant aux translations mais n'est pas invariant aux rotations, il est donc important de distinguer parmi les méthodes de déformation celles (i) ne préservant pas les rotations, (ii) préservant les rotations pour des éditions avec des transformations de faibles amplitudes et (iii) avec des transformations de grandes amplitudes ;
- les méthodes de résolution sont multiples, nous distinguons celles étant linéaires et celles non linéaires ;
- les méthodes de déformation se distinguent également par l'intégration de nouvelles contraintes établies hiérarchiquement entre elles comme par exemple celles de type position des points de contrôle, volume, squelette, etc.

**Édition simple par des contraintes de position** Lipman [109], Sorkine [156] et Yu [189] sont les premiers à utiliser l'opérateur Laplacien défini par l'équation 5.24 pour de la reconstruction de maillage. Comme décrit précédemment, la méthode consiste à préserver l'information différentielle tout en respectant des contraintes de position sur certains sommets du maillage (Équation 5.19). Toutefois, les coordonnées Laplaciennes n'étant pas invariantes aux rotations, les auteurs mettent en évidence l'intérêt d'introduire une rotation locale à chaque sommet de la géométrie pour orienter correctement les détails locaux de la géométrie. Yu [189] propose une méthode manuelle pour déterminer les rotations à appliquer et ainsi ré-orienter l'information locale. Lipman [109] et Sorkine [156] proposent une approche *automatique* basée sur l'opérateur Laplacien paramétré par des poids uniformes en introduisant un composant de rotation  $T_i$  appliqué à chaque différentielle de sommet d'indice  $i$  :

$$E(\mathbf{V}') = \sum_{i=1}^n \|T_i(\mathbf{V}')\delta_i - \mathcal{L}(\mathbf{v}')\|^2 + \sum_{i=m}^n \|\mathbf{v}'_i - \mathbf{u}_i\|^2. \quad (5.25)$$

$T_i(\mathbf{V}')$  est une transformation linéaire calculée en fonction d'une première estimation de  $\mathbf{V}'$  (à partir de l'Équation 5.19). Plus précisément,  $T_i$  approxime linéairement la rotation entre  $\mathbf{V}$  et  $\mathbf{V}'$ . Masuda [117] améliore par la suite la méthode en utilisant l'opérateur Laplacien paramétré par des poids cotangents. Ces techniques permettent finalement l'édition de surface avec des rotations de faibles amplitudes.

Zhou [194] améliore également la méthode en utilisant un graphe volumique associé à une paramétrisation Gaussienne de l'opérateur Laplacien. La méthode permet une meilleure préservation du volume et améliore les problèmes d'intersection inter-maillage mais ne préserve toujours pas complètement l'information locale de rotation lors de déformations de grandes amplitudes.

Cette limitation est directement liée à la nature linéaire de la résolution des problèmes posés. D'autres méthodes non linéaires ont donc été proposées permettant d'obtenir des formes

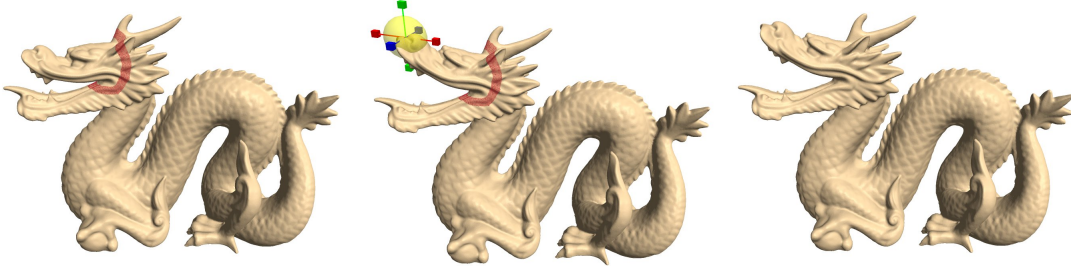


FIGURE 5.7 – Exemple d'édition de maillage interactive proposée par Sorkine [156]

naturelles pour de grandes déformations. Par exemple Krayevoy [100] introduit une nouvelle représentation simplifiée des coordonnées de la géométrie appelées *coordonnées pyramidales*. Botsch et Kobbelt [22] résolvent le problème en utilisant une représentation de la géométrie avec plusieurs niveaux de résolution. L'idée se décompose en trois phases : (i) une fonction  $f$  permet de générer une représentation simplifiée du maillage de référence sans les détails, (ii) les contraintes de position sont appliquées à la représentation simplifiée, (iii) via l'inverse de  $f$  appliqué sur le maillage simplifié on retrouve la configuration de référence intégrant les détails de la géométrie de référence. Manson [115] et Dey [48] ont également développé des méthodes inspirées de cette technique. Enfin certains auteurs [83, 115] reprennent l'équation (5.25) mais déterminent le terme  $T_i$  par une descente de gradient. L'idée est de déterminer  $T_i^k$  à partir de  $V^k$  puis  $V^{k+1}$  à partir  $T_i^k$  (puis  $T_i^{k+1}$  à partir de  $V^{k+1}$  et ainsi de suite). Pour  $k = 0$ ,  $V^k$  correspond à la première estimation de l'équation 5.19. Davantage d'informations expliquent dans l'annexe B comment sont déterminées les rotations  $T_i^k$ .

**Les techniques *As-Rigid-As-Possible*** Sorkine [159] propose une autre alternative pour éditer les maillages surfaciques impliquant de grandes déformations. L'intérêt de la technique *As-Rigid-As-Possible* (basée sur une méthode itérative) réside dans la préservation entre chaque itération de la rigidité propre à chaque cellule du maillage. En faisant l'hypothèse qu'il existe une transformation rigide entre une cellule  $C_i$  dans sa configuration de référence et après sa déformation  $C'_i$ , l'idée est de déterminer itérativement la meilleure approximation de la rotation  $R_i$  appliquée à  $C_i$  après déformation du maillage :

$$E(C_i, C'_i) = \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - R_i(\mathbf{v}_i - \mathbf{v}_j)\|^2. \quad (5.26)$$

En calculant pour chaque itération  $k$  les termes  $R_i^k$  à partir des valeurs de  $V^k$  (via une décomposition polaire), l'auteur montre que l'on peut retrouver une équation similaire à 5.19 :

$$E_a(\mathbf{V}^{k+1}) = \sum_{i=1}^n \|\delta_i^k - \mathcal{L}(\mathbf{v}_i^k)\|^2 \quad (5.27)$$

où  $\delta_i^k$  est le terme évalué à chaque itération à partir des rotations  $R_i^k$ . Une illustration des résultats obtenus par *ARAP* est donnée par la figure 5.8. La méthode est par ailleurs davantage expliquée dans l'annexe B.

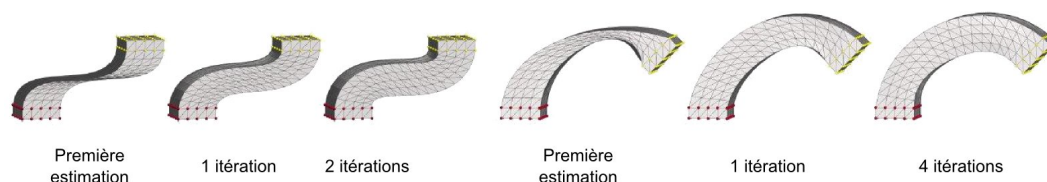


FIGURE 5.8 – Exemple d'édition de maillage utilisant *ARAP* [159]. La première estimation est le résultat de la minimisation décrite par l'équation 5.19.

Plusieurs études se sont inspirés de la technique *As-Rigid-As-Possible* pour proposer de nouvelles propriétés aux déformations. Zhang et co-auteurs [190] ont étendu l'idée précédente à des structures volumiques. Ils ont démontré l'intérêt d'introduire la notion de rigidité à l'intérieur du maillage surfacique pour préserver le volume. Borosan [21] propose une optimisation de la méthode *ARAP* appliquée à des maillages simplifiés qui servent de cage dans le cadre de déformation par les techniques *mean coordinates value* ou *harmonic coordinates*. De même Faraj et co-auteurs [54] proposent dans un problème d'édition de données volumiques d'utiliser la méthode sur des maillages cage compatibles avec *ARAP*. Les intérêts d'utiliser un maillage de contrôle (ou maillage cage) sont nombreux. Ce maillage simplifié permet une accélération des calculs et de la convergence de la méthode. Enfin, Zollhöfer [196] (figure 5.9) propose d'utiliser un espace latent basé sur la voxelisation du maillage de référence. La méthode implantée sur GPU permet de déformer des maillages quelconques et surtout d'accélérer considérablement les coûts de calcul. La méthode est toutefois limitée pour l'édition de structures complexes telles que les mains puisque la voxelisation ne respecte pas forcément les formes du maillage de référence (un voxel peut par exemple contenir deux doigts).

**Ajout de contraintes pour d'autres styles de déformation** D'autres techniques autres que celles dérivant d'*ARAP* proposent d'intégrer des contraintes de différente nature telles que la prise en compte du squelette par exemple. Huang [83] propose en 2006 une approche prenant en considération des contraintes de rigidité liées au squelette tout en préservant le volume. Les contraintes étant non linéaires, les auteurs proposent de résoudre le problème par une descente de gradient dans un sous espace simplifié (maillage de contrôle simplifié couvrant le maillage de référence). Les contraintes induites par le squelette et la préservation du volume sont projetées sur le maillage de contrôle par la méthode *interpolation de la valeur moyenne* (*mean value interpolation*). L'utilisation de ce sous espace améliore la rapidité et rend le système plus stable. Zhao améliore plus tard la méthode en utilisant un maillage de contrôle tétraédral et utilise la méthode de projection *Modified Barycentric Interpolation* ([82]). Shi [149] (figure 9.3) combine dans un même *framework* les deux approches de déformation de maillage

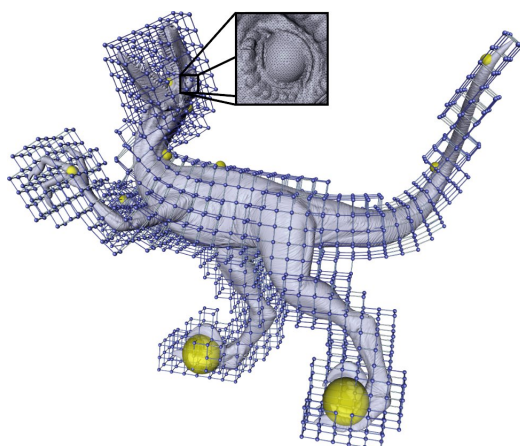


FIGURE 5.9 – Exemple d'édition de maillage utilisant ARAP [159]. La première estimation est le résultat de la minimisation décrite par l'équation 5.19.

et d'animation par *skinning*. Il définit tout d'abord une contrainte de préservation des détails de la géométrie par un représentation Laplacienne du maillage [156] puis établit le lien entre le maillage et le squelette par une phase de *rigging* classique. D'autre part il préserve certaines propriétés du squelette durant la déformation, telles que les contraintes de longueur des segments, une contrainte de basculement (le centre de gravité du modèle doit être au dessus de sa surface de sustentation) et des contraintes de distance entre certaines articulations permettant de préserver approximativement certaines butées articulaires. Son système repose également sur une descente de gradient alternant (i) une estimation de la surface de la géométrie (ii) puis à partir de celle-ci une résolution des contraintes liées au squelette à travers le lien de *rigging*.

### 5.3.1.2 Autres traitements sur les maillages

**Lissage et filtrage** Afin de lisser des maillages surfaciques, Taubin [168] est le premier auteur à avoir utilisé l'opérateur Laplacien en informatique graphique. Le principe est d'utiliser l'opérateur de Laplace-Beltrami pour atténuer localement les irrégularités en recalculant la position d'un sommet par la moyenne de ses sommets voisins. D'autres travaux ont suivi la démarche comme par exemple [47, 132, 176].

**Interpolation de maillages** Dans ce domaine deux thématiques se distinguent. (i) La première concerne l'assemblage de géométries. En mixant et interpolant les représentations différentielles de plusieurs maillages, il est possible d'assembler des géométries entre elles tout en préservant les détails d'une géométrie vers une autre. Ce principe d'interpolation entre géométries a d'abord été introduit par Alexa [2]. La technique repose sur une interpolation des coordonnées différentielles, paramétrée par un poids qui varie proportionnellement à la distance de la zone de *mélange*. Sorkine [156] (figure 5.10) améliore la méthode en ajoutant un composant

de rotation associé aux différentielles pour adapter l'orientation du maillage à transplanter sur le maillage cible. (ii) La seconde thématique concerne l'interpolation d'un maillage à un autre dont la connectivité est identique. La problématique principale réside dans la détermination des rotations à appliquer aux différentielles. Plusieurs méthodes ont ainsi été proposées pour résoudre ce problème [5, 185, 189]. Hu [81] propose une alternative plus stable en interpolant les géométries dans l'espace Laplacien dual [96].

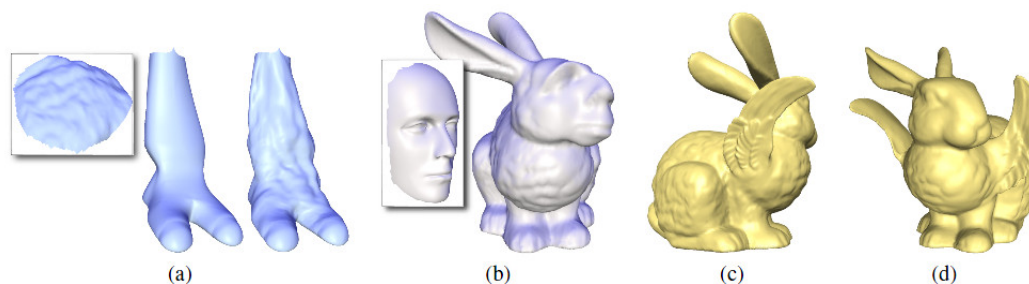


FIGURE 5.10 – Exemple de mélange de géométrie utilisant l'opérateur Laplacien [156]. (a-b) inclusion d'une géométrie dans une autre (les sources sont en médaillons). (c-d) association des ailes du modèle *Feline* avec le modèle *Bunny*.

**Description, partitionnement et mise en correspondance de graphe** L'étude spectrale de la matrice Laplacienne permet une description du graphe. En effet les valeurs propres non nulles permettent d'exprimer le graphe par l'ensemble des vecteurs propres de la matrice Laplacienne. La première valeur propre non nulle appelée valeur de *Fiedler* permet par exemple de décrire la structure principale du graphe alors que les autres valeurs propres non nulles représentent les plus hautes fréquences.

Plusieurs auteurs utilisent l'étude spectrale pour faire de la description de géométrie, e.g [141, 165], de la détection de symétrie [135] ou encore de la réduction de données sur les maillages, e.g [92]. L'idée est similaire à celle utilisée pour de la compression d'image JPEG : les composants correspondant aux hautes fréquences sont tronqués.

La description de graphe amène à d'autres applications comme le partitionnement de graphe ou la segmentation de maillage. Plus précisément, plusieurs auteurs l'utilisent pour de l'agglomération spectrale (*spectral clustering*) appliquée sur des graphes (e.g [14, 178]) ou sur des maillages (e.g [144, 154, 161]). Le problème du *clustering* consiste à partitionner un graphe en fonction du poids des arêtes en séparant les groupes entre lesquels les arêtes ont un faible poids et rassemblant celles ayant un poids de forte valeur. Plusieurs méthodes existent mais sont *NP-complet*, par exemple le clustering spectral associé à l'algorithme *K-means* permet une réduction de la complexité du problème par une solution approximative : la première valeur propre non nulle approxime bien une décomposition du graphe.

Enfin la description de graphe permet de faire de la mise en correspondance de formes

géométriques (e.g [118, 134]). Par exemple Dubrovina [51] utilise une décomposition spectrale pour établir des groupes de sommets servant de descripteurs de mise en correspondance entre une géométrie et une transformation isométrique. L'auteur combine une mesure de similarité entre les descripteurs associée à une autre entre les distances géodésiques mesurées sur les deux géométries pour établir les correspondances. L'auteur montre en particulier que les résultats dépendent fortement de la paramétrisation du Laplacien, une distance de diffusion obtient par exemple de meilleurs résultats que des distances géodésiques.

## 5.3.2 Édition de mouvement

### 5.3.2.1 Application à l'édition d'animation de maillage

L'édition d'animation de maillage consiste à répercuter l'édition spatiale d'une ou plusieurs postures de la séquence animée sur le reste du maillage ainsi que sur les autres maillages de la séquence. Tereja et Hilton [170] (figure 5.11) sont les premiers à utiliser l'opérateur Laplacien 3D discret pour éditer une posture et propage temporellement l'édition sur la séquence. Plus récemment Hétry [74] a étendu les travaux présentés dans la section 5.3.1.1 à l'utilisation de l'opérateur Laplacien discret appliqué à l'animation de maillage. Plus précisément l'auteur introduit un nouveau paramètre associé à la dimension temporelle. Ainsi l'opérateur Laplacien  $3D + t$  discret possède les mêmes propriétés que l'opérateur Laplacien 3D discret, à partir du moment où  $t$  est positif ou nul.

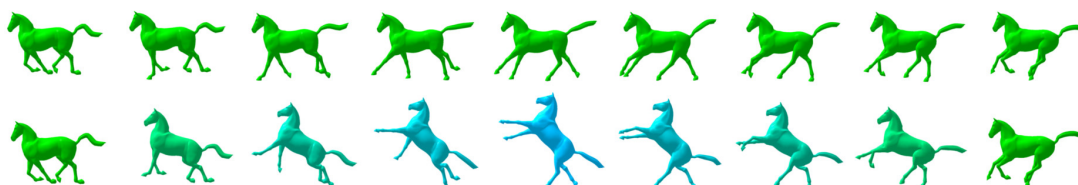


FIGURE 5.11 – Exemple d'édition d'animation de maillage spatio-temporelle sur le modèle de cheval au galop [170]. En haut la séquence originale, en bas la séquence éditée.

### 5.3.2.2 Animation de maillage guidée par des données capturées

Krayevoy & al. [100] (figure 5.12) proposent une méthode pour retrouver l'animation d'un maillage directement à partir des marqueurs sans information liée au squelette. Après une correspondance opérée manuellement entre les données capturées et le maillage, l'auteur utilise une technique basée sur une descente de gradient, appelée *Mean-value Encoding*, qui encode pour chaque sommet sa relation spatiale avec ses voisins dans un plan. Cette représentation améliore la stabilité et la vitesse de résolution de la méthode. La méthode applicable uniquement sur des géométries différentiables est efficace pour des mouvements grossiers. Elle est en

effet invariante aux rotations et permet donc des résultats réalistes pour de grandes déformations. Toutefois la méthode étant non linéaire donne des temps de calcul importants. En 2007,

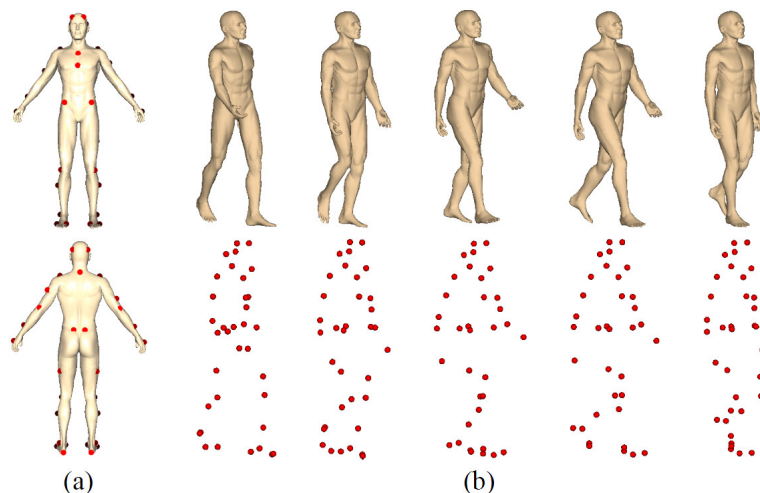


FIGURE 5.12 – Exemple de reconstruction d’animation de maillage à partir de données capturées [100]. (a) Placement des marqueurs sur le modèle. (b) (haut) Séquence animée. (en bas) données capturées originales.

Stoll et co-auteurs [163], proposent une alternative incluant des déformations avec des rotations de grandes amplitudes, en proposant un système itératif similaire à celui de Sorkine sur des maillages volumiques (formés de tétraèdres). Plus précisément, l’idée est de regrouper les sommets de chaque tétraèdre par groupe et (i) d’extraire à chaque itération la rotation  $T_j$  du tétraèdre  $j$  à partir de sa configuration courante et celle de référence ; (ii) puis de ré-injecter  $T_j$  en l’appliquant aux coordonnées différentielles dans l’Équation 5.19.

### 5.3.2.3 Application à la simulation de foule

L’opérateur Laplacien a également été utilisé dans le contexte de simulation de foule [72, 102, 167]. Ces travaux l’utilisent pour éditer interactivement les trajectoires des piétons. L’opérateur Laplacien est généralement utilisé pour encoder les relations spatiales entre les piétons et éventuellement leurs relations avec l’environnement. L’une des premières approches [102] consiste à utiliser une méthode similaire à celle vue précédemment 5.2.2.3 pour éditer les trajectoires des individus. A partir de deux configurations d’un même groupe d’individus dont la connectivité est établie par une triangulation de Delaunay, Takahashi [167] proposent une méthode d’interpolation entre les deux configurations basée sur l’étude spectrale des matrices Laplaciennes des deux groupes. Plus précisément, la décomposition spectrale donne pour chacun des groupes un ensemble de vecteur orthogonaux entre eux. Ainsi pour préserver l’orthogonalité entre les vecteurs, les auteurs reformulent le problème par un problème d’interpolation des rotations entre les deux bases propres. Dans le cas plus spécifique de l’animation de per-



sonnages articulés, Kim et co-auteurs [95] utilisent le Laplacien pour éditer les interactions entre plusieurs personnages. Les auteurs utilisent l'opérateur Laplacien pour encoder l'information spatiale et temporelle du pelvis du squelette en 2D. Le reste du corps est calculé par cinématique inverse guidée par les contacts au sol et les positions des extrémités du squelette.

#### 5.3.2.4 Application à l'édition d'animation de squelette

Ho et Komura [77] (figure 5.13) utilisent l'opérateur Laplacien discret pour éditer des séquences de squelettes. Leur méthode prend en considération les interactions entre les personnages et permet par ailleurs d'adapter un mouvement à une nouvelle morphologie ainsi qu'à une modification de l'environnement. Plus précisément la méthode utilise une structure volumique appelée *maillage interactif* encodant les relations spatiales entre le ou les personnages et l'environnement. Cette structure volumique qui inclut l'ensemble des sommets donnés par les articulations du squelette est construite à partir d'une tétraédrisation de Delaunay. Les segments des squelettes sont inclus dans les arêtes de la structure. L'édition de la structure volumique se fait de manière analogue à celle présentée dans la section 5.2.2.3. Par ailleurs, des contraintes de distances associées à la longueur des segments ainsi que des contraintes de collisions entre les os du squelette sont ajoutées. La méthode traite les postures indépendamment les unes des autres et ajoute une contrainte temporelle pour préserver la continuité sur l'ensemble de la séquence. Leur système permet un grand nombre d'édérations et permet des résultats physiquement plausibles. Toutefois la méthode est fastidieuse à mettre en oeuvre et n'est pas interactive, ce qui est dû à sa résolution non linéaire.

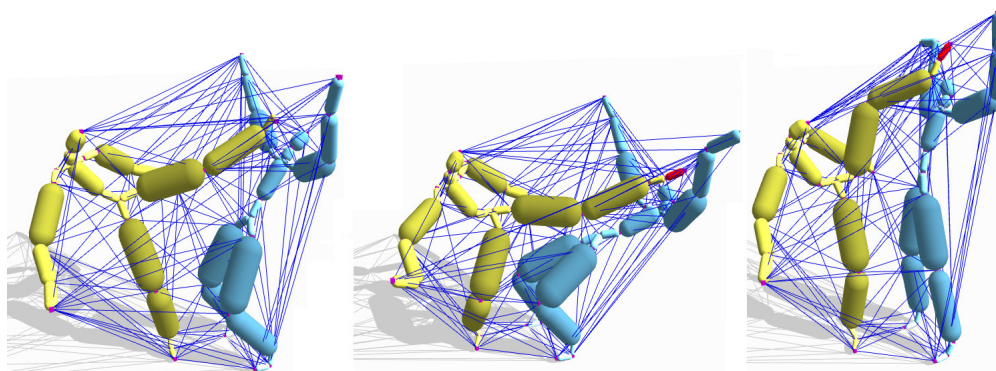


FIGURE 5.13 – Exemple d'édition d'un mouvement représentant l'interaction entre deux personnages avec différentes contraintes de position [77]. L'ensemble des segments bleu représente la structure volumique dite *maillage interactif*.

## 5.4 Conclusion

L'opérateur Laplacien possède plusieurs propriétés essentielles à l'analyse et l'édition de formes géométriques et plus généralement de graphes. La plupart des domaines d'utilisation d'un tel opérateur en informatique graphique concerne les traitements sur les maillages. Cependant, certaines études récentes appliquées aux graphes sont dédiées à l'édition de mouvement. Nous pensons en particulier à la simulation de foule ou aux travaux de Ho et Komura [77] appliqués à l'animation de squelette.

Parmi les utilisations de l'opérateur Laplacien, nous avons dans un premier temps montré comment il est possible d'éditer un graphe en fixant plusieurs points et en éditant d'autres. Dans la suite de ce document, nous étendons ce concept en le couplant avec des contraintes de distance (chapitre 6). Finalement, nous proposons un système regroupant deux types de technique dédiées à la représentation des relations spatiales dans deux espaces distincts : l'espace Laplacien et l'espace métrique. Nous verrons ensuite qu'un tel couplage est nécessaire dans le cadre de plusieurs applications dédiées au domaine du mouvement capturé. Ces différentes applications utilisent la même base de modélisation, mais introduisent une forme de spécialisation des modèles et des contraintes qui s'adapte au problème posé. Elles constituent chacune une contribution originale de cette thèse et sont décrites en détails dans les trois chapitres qui suivent.

1. Animation de personnage (chapitre 7). Pour cette application, nous présenterons une méthode capable d'éditer interactivement l'animation d'un squelette tout en préservant l'information spatio-temporelle. L'opérateur Laplacien permet de préserver les propriétés spatio-temporelles du mouvement alors que les contraintes de distance assurent le bon respect des longueurs des segments des squelettes.
2. Reconstruction de trajectoires de marqueur (chapitre 8). Nous faisons l'hypothèse dans ce chapitre qu'il existe un lien entre les trajectoires d'un marqueur avec celles de ses voisins. Nous proposons un système permettant d'exprimer ce lien par l'opérateur Laplacien associé à des contraintes préservant les distances entre certains marqueurs.
3. Complémentaire à notre deuxième application, nous proposons également un système dédié à l'animation de maillage guidée par les données issues de la capture de mouvement (chapitre 9). L'opérateur Laplacien permet dans ce contexte de donner des déformations géométriques naturelles contraintes par les données. Les contraintes de distance assurent le bon respect des longueurs des segments du squelette sous-jacent.



# 6

## Couplage des distances avec l'opérateur Laplacien

### Sommaire

---

<b>6.1 Introduction</b> . . . . .	<b>77</b>
<b>6.2 Préservation des distances</b> . . . . .	<b>78</b>
6.2.1 Présentation du problème . . . . .	78
6.2.2 Résolution par une technique de minimisation alternée . . . . .	80
<b>6.3 Application à l'édition de graphe</b> . . . . .	<b>83</b>
6.3.1 Résolution numérique . . . . .	83
6.3.2 Exemple simple . . . . .	83
<b>6.4 Conclusion</b> . . . . .	<b>83</b>

---

### 6.1 Introduction

La première partie de cette thèse nous a permis de mettre en évidence que l'utilisation des relations spatiales pour l'édition de mouvement est un principe particulièrement intéressant. Alternativement à la représentation d'un groupe de points basée uniquement sur les distances (entre ces points), nous avons vu qu'à travers l'information différentielle d'un point vis à vis de ses voisins, l'opérateur Laplacien permet une description locale des relations spatiales entre un point et son entourage. Nous avons vu que cet opérateur peut être discrétisé et ainsi appliqué à la représentation d'un graphe. Ainsi en fixant et contraignant le positionnement de certains sommets du graphe, l'opérateur Laplacien permet une déformation du graphe qui préserve au mieux les relations spatiales entre les sommets. Il en résulte des déformations visuellement naturelles et physiquement plausibles. De plus, l'intérêt d'un tel opérateur est qu'il propose une solution d'édition linéaire, ce qui présente des avantages en terme de rapidité et stabilité de la solution vis-à-vis de notre précédente étude.

Toutefois, comme nous l'avons mis en évidence dans la première partie de cette thèse, il est souvent nécessaire d'assurer la préservation de distances entre certains sommets d'un graphe.

Nous pensons en particulier aux longueurs des segments d'un squelette qui doivent rester les mêmes durant l'édition. Or l'opérateur Laplacien, bien qu'il puisse être paramétré pour donner plus ou moins d'importance à certaines relations entre sommets, n'assure pas ces contraintes de distance.

Nous proposons dans cette section une nouvelle méthode d'édition de graphe basée sur l'opérateur Laplacien incluant des contraintes de distance.

Comme nous l'avons vu dans le chapitre 3, la préservation de la distance entre deux points est un problème non linéaire. Ainsi, nous développons ici une méthode qui couple la représentation Laplacienne d'un graphe avec des contraintes de distance au moyen d'un système de minimisation alternée. Cette nouvelle méthode permet de donner un ordre de priorité entre les relations spatiales : (i) les contraintes de distance sont inviolables et donc prioritaires, elles sont assurées par une descente de gradient ; (ii) les autres relations spatiales sont d'ordre secondaire, elles sont exprimées par l'opérateur Laplacien qui suit une résolution linéaire. Ainsi, en comparaison avec la méthode présentée dans la première partie de la thèse, cette nouvelle représentation permet de relâcher fortement le nombre de contraintes non linéaires.

Dans cette partie nous rappellerons comment préserver les distances entre différents sommets d'un graphe, puis nous verrons comment inclure ces contraintes dans le problème d'édition de graphe présenté dans le chapitre 5. Enfin, nous montrerons comment mettre en œuvre un tel système et l'appliquerons sur un cas illustratif.

## 6.2 Préservation des distances

### 6.2.1 Présentation du problème

**Édition de graphe :** Pour rappel, nous considérons le graphe  $G = (\mathbf{V}, \mathbf{E})$ , où  $\mathbf{E}$  est la connectivité et  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  décrit la géométrie par les coordonnées Euclidiennes de chaque point du graphe dans  $\mathbb{R}^3$ .  $\mathbf{V}$  étant l'ensemble de  $n$  sommets du graphe  $G$  et  $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_m\}$  l'ensemble de  $m$  points de contrôles ( $m \ll n$ ), nous obtenons  $\mathbf{V}' = \{\mathbf{v}'_1, \dots, \mathbf{v}'_n\}$  le vecteur d'inconnues qui minimise la fonction quadratique :

$$E(\mathbf{V}) = E_l(\mathbf{V}, G) + E_u(\mathbf{V}, \mathbf{U}). \quad (6.1)$$

Le premier terme

$$E_l(\mathbf{V}, G) = \sum_{i=1}^n \|\delta_i - \mathcal{L}(\mathbf{v}_i)\|^2 \quad (6.2)$$

pénalise la différence entre les coordonnées différentielles après reconstruction, et le second terme

$$E_u(\mathbf{V}, \mathbf{U}) = \sum_{i=1}^m \|\mathbf{v}_i - \mathbf{u}_i\|^2 \quad (6.3)$$

pénalise le déplacement des points de contrôle. Le nouveau graphe obtenu respecte les contraintes de position des points de contrôle tout en maintenant au mieux les propriétés Laplaciennes du graphe, ainsi il en résulte une déformation naturelle du graphe.

**Préservation des longueurs des segments :** Le vecteur  $V'$  qui minimise  $E$  ne préserve pas les distances entre les sommets du graphe. Or nous aimerions que certaines distances soient invariantes à l'issue de l'édition de  $G$ . Nous ajoutons donc un nouveau terme d'énergie équivalent à celui décrit dans le chapitre 3 :

$$E_d(\mathbf{V}, G) = \sum_{e(a,b) \in E_S} (\|\mathbf{v}_a - \mathbf{v}_b\|^2 - d_{ab}^2)^2, \quad (6.4)$$

avec  $d_{ab}$  la distance désirée du segment entre les sommets  $a$  et  $b$ .

Finalement, nous pouvons ré-écrire l'équation 6.1 avec ce nouvel ensemble de contraintes :

$$E = w_l E_l + w_u E_u + w_d E_d. \quad (6.5)$$

où  $w_l$ ,  $w_u$  et  $w_d$  sont respectivement les poids associés aux trois énergies : Laplacienne, contraintes de position et de distance.

**Contraintes dures et contraintes souples** Dans le chapitre 3, où l'ensemble du système était exprimé en terme de distance, le problème était sur-contraint. Ainsi nous avons mis en évidence l'intérêt de dissocier les contraintes de natures inviolables : *les contraintes dures* de celles à préserver au mieux : *les contraintes souples*.

Ainsi posée, la résolution du problème est différente de celle préalablement proposée, puisque nous pouvons donner des priorités aux différentes contraintes via les poids  $w_l$ ,  $w_u$  et  $w_d$ . Cependant, l'objectif reste le même : il s'agit de privilégier la préservation des contraintes de distance. Ainsi, nous appelons dans la suite de ce document *contraintes dures*, les contraintes de distances, alors que les contraintes liées aux coordonnées différentielles et celles de positions sont qualifiées de *souples*. D'un point de vue technique, ce sont bien l'attribution des poids  $w_l$  et  $w_u$  donnés par l'utilisateur qui donnent l'importance aux contraintes. Le choix de ces poids est discuté dans la section expérimentation.

**Un problème non linéaire** Comme nous l'avons précédemment mentionné dans le chapitre 3, il est notable que la contrainte de distance ajoute une forte non linéarité à la fonction d'énergie  $E$ , ce qui implique une minimisation dépendante de minima locaux et conduit parfois à la non convergence de l'algorithme. Nous proposons dans la section suivante une méthode originale et efficace pour résoudre ce problème.

## 6.2.2 Résolution par une technique de minimisation alternée

La plupart des études associant les propriétés Laplaciennes avec des contraintes de distance [78, 180] combinent dans un système itératif et global des contraintes linéaires et non linéaires. Ici nous présentons un algorithme de minimisation alternée qui découple la résolution de  $E$  en deux parties linéaire et non linéaire.

### 6.2.2.1 Relaxation des contraintes de distance

Dans un premier temps, nous relaxons la contrainte de distance en exprimant l'équation (6.4) par un système linéaire équivalent approché.

**Exemple pour deux points** En considérant les vecteurs différentiels  $\gamma_{ab}$  et  $\gamma_{ba}$  entre les sommets  $a$  et  $b$ , nous pouvons observer que  $\gamma_{ab} = -\gamma_{ba} = \mathbf{v}_a - \mathbf{v}_b$  et que leurs normes  $\|\gamma_{ab}\| = \|\gamma_{ba}\| = d_{ab}$ . Comme la figure 6.1 l'illustre (dans le plan), notre idée est de préserver la norme du vecteur  $\gamma_{ab}$  (et donc de  $\gamma_{ba}$ ) le long de la droite  $D$ .

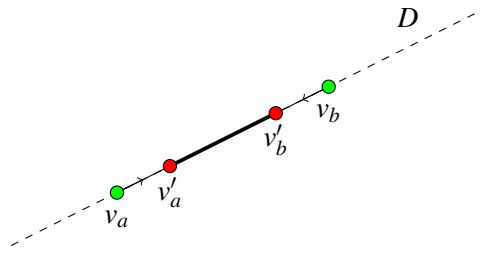


FIGURE 6.1 – Application de la contrainte de distance sur les sommets  $\mathbf{v}_a$  et  $\mathbf{v}_b$  le long d'une droite  $D$ .  $\mathbf{v}'_a$  et  $\mathbf{v}'_b$  sont les sommets tels que  $\|\mathbf{v}'_a - \mathbf{v}'_b\| = d_{ab}$ .

En considérant les coordonnées en  $x$ ,  $y$  et  $z$  des sommets  $a$  et  $b$  par  $\mathbf{v}_{a,x}$ ,  $\mathbf{v}_{a,y}$  et  $\mathbf{v}_{a,z}$  et  $\mathbf{v}_{b,x}$ ,  $\mathbf{v}_{b,y}$  et  $\mathbf{v}_{b,z}$ , la linéarisation du problème se traduit par un système linéaire :

$$\begin{aligned} \mathbf{v}_{a,x} - \mathbf{v}_{b,x} &= \gamma_{ab,x} \\ \mathbf{v}_{a,y} - \mathbf{v}_{b,y} &= \gamma_{ab,y} \\ \mathbf{v}_{a,z} - \mathbf{v}_{b,z} &= \gamma_{ab,z}, \end{aligned} \tag{6.6}$$

équivalent à

$$\begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_{a,x} & \mathbf{v}_{a,y} & \mathbf{v}_{a,z} \\ \mathbf{v}_{b,x} & \mathbf{v}_{b,y} & \mathbf{v}_{b,z} \end{pmatrix} = \begin{pmatrix} \gamma_{ab,x} & \gamma_{ab,y} & \gamma_{ab,z} \end{pmatrix}. \tag{6.7}$$

**Application à l'ensemble du système** Ainsi, nous pouvons généraliser ce concept pour l'ensemble des contraintes de distance. A partir de  $E_S$  l'ensemble des arêtes de  $G$  de longueurs

fixes, nous définissons la contrainte  $E_d$  des distances invariantes associées à ces arêtes par la fonction quadratique  $E_d$  :

$$E_d(\mathbf{V}', \Gamma) = \sum_{e(i,j) \in E_S} \|\gamma_{ij} - \mathcal{D}_{ij}(\mathbf{V}')\|^2, \quad (6.8)$$

où  $\mathcal{D}_{ij}(\mathbf{V}') = \mathbf{v}'_i - \mathbf{v}'_j$ , et  $\Gamma$  est un vecteur qui stocke l'ensemble des valeurs possibles des  $\gamma_{ij}$ . Minimiser cette fonction d'énergie permet de faire correspondre en terme de norme et d'alignement de vecteurs  $\mathbf{v}'_i - \mathbf{v}'_j$  et  $\gamma_{ij}$ . Résoudre l'équation (6.8) revient à résoudre le système linéaire suivant au sens des moindres carrés :

$$D\mathbf{V}' = \Gamma \quad (6.9)$$

où  $D$  est la matrice de dimension  $\text{card}(E_S) \times n$  dont le rôle est de calculer les coordonnées différentielles de chaque segments  $\in E_S$ .

Finalement, la fonctionnelle  $E_d$  garantie un bon respect des contraintes de distance mais nécessite le calcul des  $\gamma_{ij}$ . Nous voyons par la suite comment déterminer le vecteur de coordonnées différentielles  $\Gamma$ .

### 6.2.2.2 Détermination de $\Gamma$

La fonction d'énergie  $E$  devient finalement :

$$E = w_l E_l + w_u E_u + w_d E_d. \quad (6.10)$$

Le dernier terme  $E_d$  est maintenant linéaire mais dépend de  $\Gamma$  qui reste à déterminer.  $\Gamma$  est en fait la solution de l'équation (6.4) considérée seule (puisque  $\gamma_{ab} = \mathbf{v}_a - \mathbf{v}_b$ ), mais dont l'initialisation est fournie par une première estimation du problème sans les contraintes de distance.

Plus précisément, nous utilisons la méthode décrite dans la première partie de ce manuscrit (chapitre 3) pour faire correspondre les contraintes de distance avec  $\mathbf{V}'$ . Le problème étant non linéaire, nous utilisons la descente de gradient de Newton Raphson pour trouver  $\mathbf{V}'$ . Le pré-conditionnement de l'algorithme de Newton Raphson est donné par une première estimation de  $\mathbf{V}'$  calculée par l'équation (6.1) (déjà proche de la solution).

### 6.2.2.3 Synthèse du problème

**Forme matricielle de  $E$**  L'équation  $E$  (6.10) peut être mise sous la forme matricielle :

$$\begin{pmatrix} \mathbf{L} \\ w_u \mathbf{I}_U \\ w_d \mathbf{D} \end{pmatrix} \mathbf{V}' = \begin{pmatrix} \Delta_R \\ w_u \mathbf{U} \\ w_d \Gamma \end{pmatrix} \quad (6.11)$$



Notons  $A$  la matrice du premier terme de cette équation. Pour clarifier la construction de  $A$ , nous proposons à travers la figure 6.2 d'illustrer sa construction à partir d'un exemple simple.  $A$  concatène les trois matrices Laplaciennes, de contraintes de position et de contraintes de distance :

- les informations de voisinage définies par les segments noirs (entre les sommets) permettent de construire la matrice Laplacienne  $L$  en bleu dans  $A$ .
- Le sommet  $v_0$  sert d'ancre et nous lui associons une contrainte de position définie en jaune dans  $A$ .
- Enfin les pointillés rouges représentent les contraintes de distance. Elles sont décrites par la matrice rouge de  $A$ .

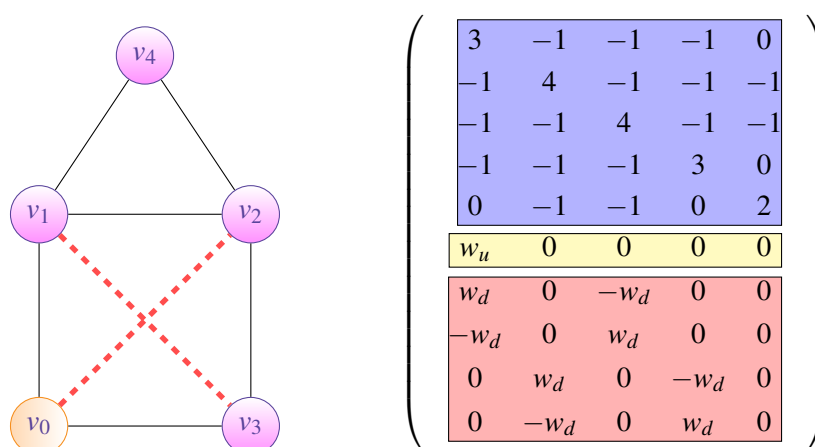


FIGURE 6.2 – Un exemple de graphe et sa matrice correspondant au premier terme du système 6.11 associé. Sur le graphe en jaune le sommet fixe et en pointillés rouges les arêtes de longueurs fixes. En jaune et en rouge dans la matrice finale, leurs sous matrices de contraintes associées.

**Processus** Finalement notre méthode décrite par l'algorithme 1 suit le processus suivant :

1. L'initialisation du système se fait par une première estimation de  $V'$  calculée par l'équation (6.1).  $V'$  est proche d'une configuration acceptable mais ne respecte pas les contraintes de distance.
2. Nous soumettons  $V'$  aux contraintes de distance définies par l'équation (6.4).
3. Les nouvelles valeurs de  $V'$  respectent les contraintes de distance ce qui nous permet d'extraire les valeurs de  $\Gamma$ .  $\Gamma$  est ensuite appliqué au système  $E$  décrit par l'équation (6.10).

L'addition des contraintes dans  $E$  implique que notre système linéaire est sur-déterminé, la solution est donc approximative après une itération. Ce système n'assure pas complètement que les contraintes de distance soient respectées. Nous ré-appliquons donc le processus des étapes 2

et 3 jusqu'à convergence. Le terme  $E_d(\mathbf{V}', \Gamma)$  n'étant pas orthogonal d'une itération sur l'autre, la convergence est assurée. Par ailleurs, les poids  $w_u$  et  $w_d$  définis par l'utilisateur permettent de donner plus ou moins d'importance aux contraintes. Dans notre cas, les contraintes de distance étant plus importantes que les autres, la relation  $w_d > w_u$  garantit un bon résultat. Finalement, en pratique seulement quelques itérations sont requises pour résoudre le système.

**Données:**

$\mathbf{V}$  : Vecteur de sommets du graphe

$\mathbf{V}'$  : Vecteur des nouvelles positions après édition

$\mathbf{U}$  : Un ensemble de contraintes de position

$G$  : le graphe original

1  $\mathbf{V}' = \operatorname{argmin}_{\mathbf{V}} E_l(\mathbf{V}) + E_u(\mathbf{V})$

2 // Contraintes Laplacienne et de position ;

3 **tant que pas de convergence faire**

4     Calcul de  $\hat{\Gamma}$  à partir de  $\operatorname{argmin}_{\mathbf{V}} E_d(\mathbf{V}, G)$  (équation 6.4), pré-conditionné avec  $\mathbf{V}'$  ;

5     
$$\mathbf{V}' = \operatorname{argmin}_{\mathbf{V}} \begin{pmatrix} w_l \mathbf{L} \\ w_u \mathbf{I}_U \\ w_d \mathbf{D} \end{pmatrix} \mathbf{V} = \begin{pmatrix} \Delta \mathbf{R} \\ w_u \mathbf{U} \\ w_d \hat{\Gamma} \end{pmatrix}$$

**Algorithm 1:** ALGORITHME D'ÉDITION DE GRAPHE

## 6.3 Application à l'édition de graphe

### 6.3.1 Résolution numérique

Le système étant de rang plein, il a une solution unique au sens des moindres carrés. Nous utilisons donc une factorisation de Cholesky pour résoudre le vecteur solution. De plus la matrice du premier membre du système (6.11) étant fortement éparsée, nous utilisons une décomposition de Cholesky sur matrice creuse, afin d'optimiser fortement les performances, notamment pour du temps réel.

### 6.3.2 Exemple simple

L'exemple illustré par la figure 6.3 montre le succès de notre méthode. Par ailleurs, nous pouvons observer l'intérêt de l'ajout des contraintes de distance qui permettent de donner un style à la déformation Laplacienne.

## 6.4 Conclusion

De même que la méthode développée dans le chapitre 3, cette méthode permet de caractériser un sommet par sa relation spatiale aux autres sommets du graphe. Cependant, cette

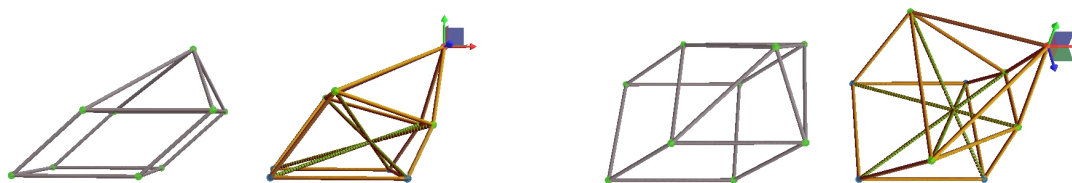


FIGURE 6.3 – Déformation simple. vue de profil(à droite) et du dessus(à gauche) de la déformation Laplacienne (en gris) et Laplacienne contrainte par les distance (en jaune). Les segments en pointillés verts représentent les contraintes de distance invariantes durant la déformation.

nouvelle proposition permet de relâcher fortement la complexité du premier problème posé puisque seule une sous partie des arêtes est soumise à des contraintes de distance et donc à une résolution non linéaire.

Afin de valider notre méthode, nous proposons dans la suite de ce document de l'appliquer à trois cas d'étude distinctes sur trois types de représentation du mouvement à partir de données capturées :

- l'opérateur Laplacien 3D+t pour l'édition de mouvement basée squelette (chapitre 7). Il s'agit d'utiliser l'opérateur Laplacien sur une représentation originale du mouvement encodant dans une structure unifiée l'information spatiale et temporelle. Les contraintes de distances servent à préserver les longueurs des segments des squelettes ;
- l'opérateur Laplacien pour la reconstruction de trajectoires multivariés (chapitre 8). Nous utilisons dans ce contexte le couplage de l'opérateur Laplacien avec les contraintes de distance pour reconstruire les trajectoires manquantes dans l'espace Euclidien. L'information contenue par les coordonnées différentielles permet de retrouver sa position et donc les trajectoires manquantes. Les contraintes de distance permettent de préserver les distances entre certains marqueurs durant la reconstruction. Nous appliquons notre système à la reconstruction du mouvement des marqueurs issu de la capture de mouvement ;
- l'opérateur Laplacien pour l'animation de maillage guidée par les trajectoires des marqueurs issues de la capture de mouvement (chapitre 9). Dans ce contexte, l'opérateur Laplacien permet de préserver les propriétés géométriques du maillage. Les contraintes de distance servent à préserver les longueurs des segments du squelette sous-jacent.

# 7

## L'opérateur Laplacien 3D+t pour une édition spatio-temporelle du mouvement

### Sommaire

---

<b>7.1</b>	<b>Introduction</b>	<b>85</b>
<b>7.2</b>	<b>Approches connexes</b>	<b>88</b>
<b>7.3</b>	<b>Utilisation de l'opérateur 3D+t pour l'édition de mouvement</b>	<b>88</b>
7.3.1	L'opérateur discret Laplacien 3D+t	88
7.3.2	Du mouvement capturé vers le graphe 3D+t	89
7.3.3	Couplage des distances avec l'opérateur Laplacien pour l'édition du mouvement	89
7.3.4	Poids Laplacien	91
7.3.5	Contraintes de rotation	92
<b>7.4</b>	<b>Application à l'édition de mouvement</b>	<b>93</b>
7.4.1	Édition de mouvement	93
7.4.2	Adaptation de mouvement	96
<b>7.5</b>	<b>Discussion et perspectives</b>	<b>97</b>

---

### 7.1 Introduction

L'édition de mouvement humain est un sujet délicat en animation de personnage. En effet, ce type d'opération requiert des données de qualité ainsi qu'une connaissance approfondie des caractéristiques essentielles du mouvement. La technique classique pour éditer et créer de nouveaux mouvements revient à spécifier manuellement de nouveaux objectifs et contraintes, tout en préservant le plus possible les propriétés spatio-temporelles, l'expressivité et la cohérence du mouvement original.

Durant la décennie passée, plusieurs études se sont focalisées sur les problèmes d'adaptation, de déformation et d'édition interactive du mouvement à partir de mouvements existants. La figure 7.1 illustre deux types d'édition de mouvement classiques : modification spatiale

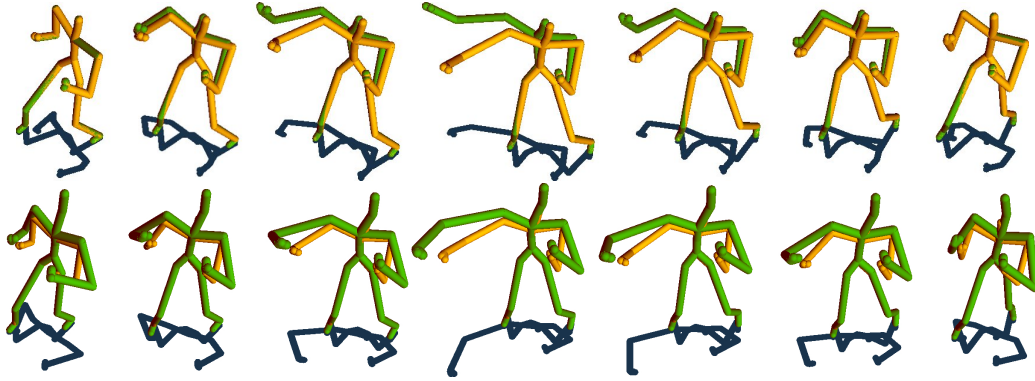


FIGURE 7.1 – Exemple d'édition de mouvement basé squelette. Edition de mouvement avec un nouvel objectif (en haut), Adaptation de mouvement pour un redimensionnement du squelette (en bas). En jaune le mouvement de référence, en vert le nouveau mouvement.

d'une caractéristique du mouvement (en haut) et redimensionnement du mouvement pour un squelette d'une autre morphologie (en bas). Dans les deux cas, l'information temporelle est à préserver. Parmi les approches prometteuses, certaines ont mis en évidence l'intérêt d'utiliser les coordonnées Laplaciennes pour encoder implicitement les relations spatiales des différents éléments du squelette avec lui même ou avec d'autres personnages. Comme nous l'avons vu précédemment, ces relations spatiales sont particulièrement intuitives et appropriées pour décrire la sémantique des interactions. Ce concept a été appliqué avec succès pour de l'édition et de l'adaptation de mouvements capturés [78]. Cependant, si les méthodes basées sur l'édition Laplacienne permettent des déformations naturelles et préservent au mieux les relations spatiales, elles ne traitent pas de l'utilisation de l'opérateur Laplacien pour la préservation de la dynamique du mouvement.

Nous proposons dans ce chapitre, une nouvelle méthode d'édition de mouvement basée sur une utilisation de l'opérateur Laplacien couplant dans un même traitement l'information temporelle et spatiale. Cela nous permet de préserver automatiquement les déformations géométriques et la dimension temporelle du mouvement.

Pour ce faire, nous étendons l'expression classique de l'opérateur Laplacien pour de l'animation de squelette en incorporant le temps  $t$  couplé aux coordonnées Laplaciennes 3D  $x$ ,  $y$  et  $z$ . Appliqué à une fonction  $f$ , l'opérateur Laplacien devient :

$$\Delta f = \frac{\partial^2 f}{\partial t^2} + \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} .$$

Dans cette équation, la dérivée seconde évalue l'accélération du mouvement dans le voisinage temporel de la posture courante. L'intérêt d'introduire le temps dans l'édition est de préserver localement l'accélération et la décélération, la dérivée comportant une information sur la dynamique du mouvement.

Dans cette section, le mouvement est représenté par une structure géométrique globale - appelée le graphe 3D+t - qui encode dans le même temps l'information spatiale par une

connectivité établie suivant les segments des squelettes, et l'information temporelle qui représente les trajectoires des articulations dans le temps. Le processus d'édition est donc obtenu en utilisant la représentation Laplacienne du graphe 3D+t. Finalement, en fixant certains points du graphe manuellement et en préservant la bonne longueur des segments des squelettes, nous parvenons via l'opérateur Laplacien à éditer significativement le mouvement tout en préservant les propriétés spatio-temporelles du mouvement d'origine.

Le nouveau mouvement créé suit notre méthode décrite dans le chapitre 6. Ainsi il est le fruit d'une double minimisation linéaire et non linéaire appliquée au graphe 3D+t. Rappelons que la première minimisation pénalise la différence d'énergie entre les coordonnées Laplaciennes du mouvement d'origine et du mouvement désiré, et la seconde, pré-conditionnée par la première minimisation, utilise une descente de gradient ayant pour objectif de préserver les longueurs des segments. Par ailleurs, bien que nos travaux se différencient par la préservation de contraintes de distance, notre méthode est similaire à l'approche développée par [74] appliquée à de l'animation de maillage. Plus précisément, notre approche permet de préserver implicitement i) les relations spatiales, données par la connectivité induite par les segments des squelettes et les distances aux différents objets de l'environnement, et ii) les relations temporelles à travers la connexion entre les articulations d'une posture à l'autre sur l'ensemble du mouvement.

Finalement nos contributions sont les suivantes :

- nous proposons une méthode unique pour coupler dans un même temps les variations temporelles et spatiales du mouvement à travers le graphe 3D+t. Associé à l'opérateur Laplacien, notre système permet de préserver durant le processus d'édition les propriétés spatiales et temporelles du mouvement d'origine ;
- au lieu d'utiliser les poids uniformes associés aux arêtes du graphe, nous introduisons une paramétrisation des coordonnées Laplaciennes par des poids suivant un noyau Gaussien ce qui permet de paramétrer la dynamique temporelle ;
- un contrôle des rotations est directement extrait à partir des points de contrôle et utilisé dans la fonction de minimisation pour modifier efficacement la séquence de squelettes via un processus d'interpolation ;
- nous utilisons le schéma d'optimisation numérique proposé dans le chapitre 6 pour découpler dans un processus itératif et non linéaire les différentes minimisations.

Dans cette section, nous développons un système complet, robuste et interactif pour éditer des mouvements capturés. Notre système est capable d'éditer tout type de mouvement (marche, sauts, boxe, natation, etc), en permettant un contrôle intuitif de l'utilisateur. De plus, comme notre système préserve les proximités et les contacts en maintenant la distance entre les points 3D, il est aussi approprié pour l'adaptation morphologique de mouvement.

## 7.2 Approches connexes

Il existe un grand nombre d'études dédiées à l'édition de mouvement. Cette partie du manuscrit étant naturellement liée au chapitre 3, le lecteur qui s'intéresse aux approches connexes concernant l'édition de mouvement peut trouver davantage d'informations dans la section 3.2. Il est intéressant toutefois de différencier les deux approches. Notre première contribution utilise l'espace métrique pour représenter le mouvement et est dédiée à de la génération de mouvement : création d'un nouveau mouvement à partir d'un objectif à atteindre. Notre seconde approche se focalise davantage sur de la déformation de mouvement : modification d'un mouvement existant. Nous rappelons les travaux les plus importants liés à cette approche.

La plupart du temps, le mouvement existant est édité pour produire une nouvelle animation en utilisant des techniques de cinématique inverse (e.g. [29, 60, 69]) ou en construisant un modèle statistique du mouvement pour en contraindre la synthèse [33, 43, 84]. [43] propose de produire aléatoirement de nouvelles trajectoires en s'inspirant des informations qui précèdent et succèdent la partie d'animation à éditer. Dans la majorité des cas, ces méthodes statistiques nécessitent une importante quantité de données pour apprendre un bon modèle de mouvement. Ce n'est pas le cas avec de la cinématique inverse qui présente l'inconvénient de perdre une partie de la qualité du signal d'origine quand les nouveaux objectifs imposent une forte différence avec le mouvement de référence. Les modèles physiques e.g [166] sont des alternatives intéressantes pour modifier le mouvement original soumis à des contraintes physiques, mais sont plus coûteuses en temps de calcul et peuvent conduire à des instabilités. Dans notre système, les caractéristiques des mouvements originaux sont préservées en changeant l'espace de représentation du mouvement par une représentation différentielle basée sur des coordonnées Laplaciennes.

## 7.3 Utilisation de l'opérateur 3D+t pour l'édition de mouvement

Dans cette section, notre objectif est de fournir un système d'édition qui préserve dans un même temps les variations spatiales et temporelles du mouvement. Pour ce faire, nous utilisons l'opérateur Laplacien qui calcule les coordonnées différentielles de chaque sommet du graphe avec ses voisins dans l'espace et dans le temps. Ainsi, nous étendons les précédentes présentations de l'opérateur Laplacien discret 5.2.2 et 6 pour minimiser localement la déformation spatiale et l'énergie d'accélération.

### 7.3.1 L'opérateur discret Laplacien 3D+t

Soit une séquence de squelettes  $S = (S_1, \dots, S_m)$ ; chaque squelette  $S_k$  ( $1 \leq k \leq m$ ) étant défini par  $S_k = (\mathbf{V}_k, \mathbf{E}_k)$ , avec  $\mathbf{V}_k$  l'ensemble de sommets de  $S_k$  et  $\mathbf{E}_k$  l'ensemble d'arêtes. Nous notons  $v_{i,k}$ , le sommet d'index  $i$  contenu dans  $\mathbf{V}_k$ . Comme il est décrit dans [74], nous définissons les coordonnées Laplaciennes 3D+t du sommet  $v_{i,k}$ , comme la somme des coordon-

nées spatiales correspondant au squelette  $k$  et des coordonnées temporelles correspondant aux arêtes "temporelles"  $(\mathbf{v}_{i,k-1}, \mathbf{v}_{i,k})$  et  $(\mathbf{v}_{i,k}, \mathbf{v}_{i,k+1})$ . L'ensemble de ces coordonnées Laplaciennes peuvent être associées à des poids spécifiques :

$$\mathcal{L}(\mathbf{v}_i) = (w^-(\mathbf{v}_{i,k} - \mathbf{v}_{i,k-1}) + w^+(\mathbf{v}_{i,k} - \mathbf{v}_{i,k+1})) + \sum_{j \in \mathcal{N}_k(i)} w_{ij,k}(\mathbf{v}_{i,k} - \mathbf{v}_{j,k}), \quad (7.1)$$

où  $w^-$  et  $w^+$  sont les poids de chaque arête "temporelle" liés à  $\mathbf{v}_{i,k}$ , et  $\mathcal{N}_k(i)$  est l'ensemble des voisins de  $\mathbf{v}_{i,k}$ . Nous détaillerons la détermination de ces poids dans la section 7.3.4. De plus, pour simplifier les notations, nous ne distinguons pas (au sens de la notation) dans la suite de ce document les poids caractérisants les arêtes temporelles et spatiales.

### 7.3.2 Du mouvement capturé vers le graphe 3D+t

Dans un premier temps, notre méthode convertit la séquence de squelettes dans le temps en une structure géométrique : le graphe 3D+t. Il encode à la fois les données spatiales (articulations du squelette) et les données temporelles (liens entre les squelettes adjacents). Comme illustré dans la figure 7.2, nous créons le graphe 3D+t en extrayant à partir du mouvement original la sous séquence que nous désirons éditer. Il n'est pas nécessaire d'appliquer une déformation sur l'ensemble de la séquence puisque le temps de calcul est directement lié à la taille du graphe (et donc du mouvement à extraire). L'information géométrique est donnée par l'ensemble des positions des articulations de tous les squelettes à extraire. Comme nous l'avons mentionné précédemment, l'information topologique du graphe est déterminée par la connectivité de chaque squelette (une arête correspond à un segment), associée aux liens entre les squelettes. Ainsi, comme illustré dans la figure 7.2, le sommet  $\mathbf{v}_{i,k}$  de l'articulation  $i$  du squelette au temps  $k$  est lié aux sommets  $\mathbf{v}_{i,k-1}$  et  $\mathbf{v}_{i,k+1}$  correspondant à l'articulation  $i$  des squelettes aux temps  $k-1$  et  $k+1$ .

Nous faisons ici le lien avec notre étude développée dans la section 6 du chapitre 6. Ainsi, nous pouvons redéfinir le graphe 3D+t par  $G = (\mathbf{V}, \mathbf{E}) = (\mathbf{V}, \mathbf{E}_S \cup \mathbf{E}_T)$ , avec  $\mathbf{V} = \{\mathbf{v}_k\}$  l'ensemble des points,  $\mathbf{E}$  l'ensemble des arêtes,  $\mathbf{E}_S$  l'ensemble des arêtes correspondant aux segments des squelettes, et  $\mathbf{E}_T$  l'ensemble des arêtes temporelles entre les squelettes adjacents dans le temps. Notons que  $\mathbf{E}_T$  correspond au premier terme de l'équation (7.1) et  $\mathbf{E}_S$  au second terme. Enfin rappelons que nous notons  $e(i, j)$  l'arête qui relie les points d'index  $i$  et  $j$ .

### 7.3.3 Couplage des distances avec l'opérateur Laplacien pour l'édition du mouvement

Notre méthode d'édition repose sur un déplacement manuel des points de contrôle, que ce soit dans l'espace ou dans le temps, tout en préservant les longueurs des segments des squelettes durant l'édition. Notons que le graphe  $G$  présenté dans ce chapitre est similaire à celui présenté



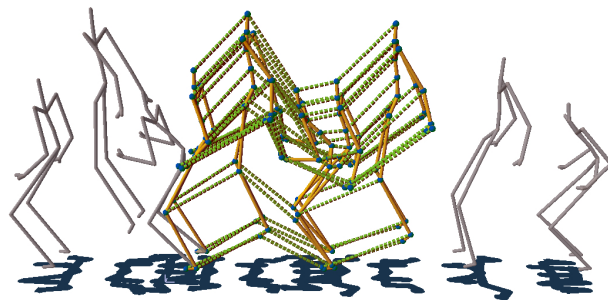


FIGURE 7.2 – Illustration de l'extraction et de la construction de  $G$ . Les postures de référence sont représentées en gris. Les points bleus représentent l'information géométrique. La connectivité du graphe est représentée en pointillé et jaune. Les arêtes jaunes sont extraites à partir des segments des squelettes et en pointillés vert, les nouvelles arêtes créées à partir des articulations (sommets bleus) à partir d'une posture vers une autre.

dans la section 6 du chapitre 6 et possède donc les mêmes propriétés. Nous expliquons ici comment lui appliquer les méthodes précédemment décrites pour éditer le mouvement.

Rappelons que nous avons proposé une méthode d'édition de graphe qui préserve un ensemble de distances en minimisant la fonction quadratique  $E$  :

$$E = w_l E_l + w_u E_u + w_d E_d. \quad (7.2)$$

où  $E_l$  pénalise la différence entre les coordonnées différentielles après reconstruction,  $E_p$  pénalise le déplacement des points de contrôle. Appliquée à notre contexte  $E_d$  minimise la différence de distance correspondant aux longueurs des segments de l'ensemble des squelettes.  $w_l$ ,  $w_u$  et  $w_d$  sont respectivement les poids associés aux coordonnées Laplaciennes, contraintes de position et contraintes de distance.

La figure 7.3 illustre le résultat de notre méthode sur un mouvement de boxe. Les postures initiale, éditée et finale sont données en haut de la figure. En bas, nous montrons le résultat de l'édition du graphe associé et le nouveau mouvement (en rouge). La posture produite est cohérente, le nouveau graphe respecte :

- les contraintes de distance correspondant aux longueurs des segments (l'erreur est quasi nulle pour une dizaine d'itérations) ;
- les positions des points de contrôle représentés dans le repère de la cible ;
- préserve les propriétés Laplaciennes du graphe.

Il en résulte ainsi une déformation naturelle du graphe. Cependant, cette première approche donne lieu à plusieurs inconvénients :

- les poids  $w_{ij}$  définis par les équations (5.10) et (7.1) influencent fortement le résultat de l'optimisation et laissent apparaître des comportements incohérents sur certaines manipulations concernant la dynamique du mouvement ;
- le graphe déformé paraît cohérent tant que les rotations sont de faibles amplitudes.

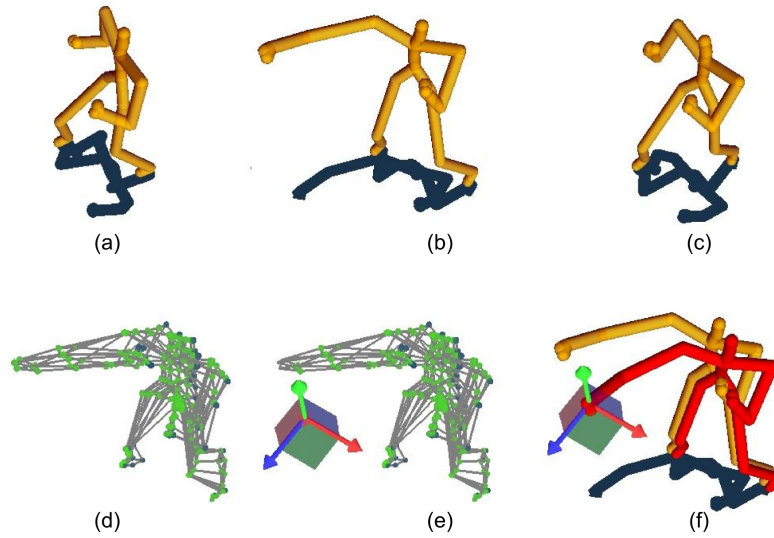


FIGURE 7.3 – En haut : (a) posture initiale du sous-mouvement, (b) posture sur laquelle est opérée l'édition du sous mouvement, (c) posture finale du sous-mouvement. En bas : (d) le graphe 3D+t généré à partir d'un sous mouvement de boxe, en bleu les points fixes de référence de l'animation source, (e) en rouge les nouvelles positions des points "cibles"; (f) le mouvement calculé : en jaune le squelette de l'animation source au moment de la frappe. En rouge le nouveau mouvement calculé.

Nous proposons de considérer ces deux problèmes dans les prochains paragraphes.

### 7.3.4 Poids Laplacien

Comme énoncé précédemment, pour de l'édition de maillage, l'opérateur Laplacien est généralement paramétré par des poids uniformes [156] ou des poids cotangents [122]. Dans notre méthode, la nature duale espace/temps du graphe induit naturellement de considérer différemment les arêtes temporelles de celles liées à l'espace. Paramétrer l'opérateur Laplacien avec des poids uniformes n'est pas approprié, puisque il en résulte un lissage du graphe édité. Nous introduisons donc une paramétrisation exponentielle de la matrice Laplacienne pour donner plus d'importance à certaines arêtes du graphe vis-à-vis des autres. Cette notion est en particulier directement liée à la notion de proximité *temporelle* entre les sommets du graphe. Ainsi, les poids  $w_{ij}$  sont définis par :

$$w_{ij} = \begin{cases} 1 & (i, j) \in E_S \\ 1 + \alpha e^{(-\beta \frac{d_{ij}^2}{d_{max}^2})} & (i, j) \in E_T \\ 0 & \text{sinon} \end{cases} \quad (7.3)$$

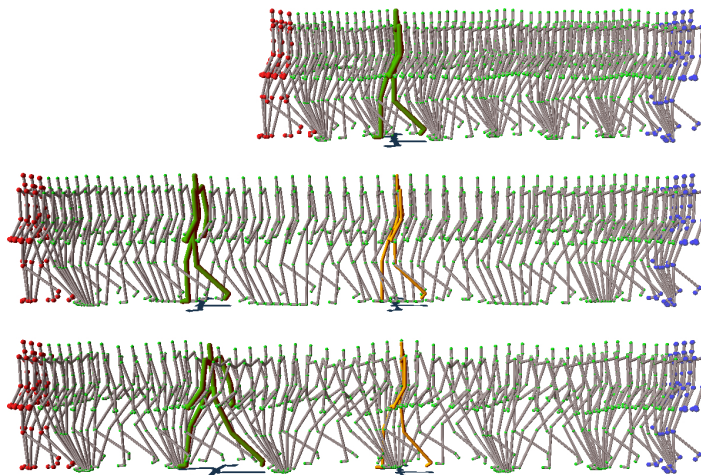


FIGURE 7.4 – Exemple de paramétrisation du graphe avec différents poids durant la construction de la matrice Laplacienne pour un mouvement de marche : en haut le mouvement d'origine sans déformation, au milieu avec des poids uniformes, et en bas avec des poids suivant une paramétrisation exponentielle. Les points bleus et rouges représentent les points de contrôle du graphe et en vert ceux représentant les positions des articulations. En jaune, le squelette de l'animation originale, et en vert l'animation résultante après déformation.

où  $\alpha$  et  $\beta$  sont des coefficients donnés par l'utilisateur.  $\alpha$  est utilisé pour définir l'importance des arêtes *temporelles* en comparaison de celles du reste du graphe, et  $\beta$  est utilisé pour paramétrer l'intensité de l'exponentielle.  $d_{ij}$  est la distance Euclidienne correspondant à l'arête entre les sommets  $i$  et  $j$ , et  $d_{max}$  est la taille maximum des arêtes dans  $\mathbf{E}_T$ . La figure 7.4 montre l'intérêt d'introduire un tel poids, en comparant le mouvement de référence (en haut de la figure), celui avec des poids uniformes (au milieu), et celui où nous avons introduit le nouveau poids  $w_{ij}$ . Nous observons une meilleure préservation de la dynamique concernant les pieds, et en particulier une forte limitation de l'effet de glissement.

### 7.3.5 Contraintes de rotation

Les coordonnées Laplaciennes sont invariantes aux translations, mais sensibles aux transformations linéaires. Cela signifie que la structure du graphe peut être translatée, mais nous ne pouvons pas appliquer de rotations. Comme le montre la figure 7.5, il est essentiel d'introduire un contrôle des rotations, puisque l'absence de rotation nuit à la déformation du mouvement qui doit pouvoir comporter des rotations globales ou locales. Nous proposons donc d'appliquer une rotation à chaque groupe de points représentant un squelette. Cette rotation est calculée à partir des orientations du premier et du dernier squelette (celui édité par l'utilisateur). Nous introduisons ainsi un nouveau terme de rotation  $T_k$  (matrice de dimension  $3 \times 3$ ), qui est appliqué aux différentiels des sommets  $\delta_i$ . L'indice  $k$  correspond à la  $k^{\text{ième}}$  posture de la séquence animée :

$$E_l(\mathbf{V}, G) = \sum_{i=1}^n \|\mathbf{T}_k \delta_i - \mathcal{L}(\mathbf{v}_i)\|^2. \quad (7.4)$$

Les rotations sont induites par les points de contrôle qui sont directement appliqués aux différents squelettes du graphe. Ainsi, pour calculer  $\mathbf{T}_k$ , nous utilisons une interpolation linéaire sphérique (avec des quaternions) entre les squelettes.

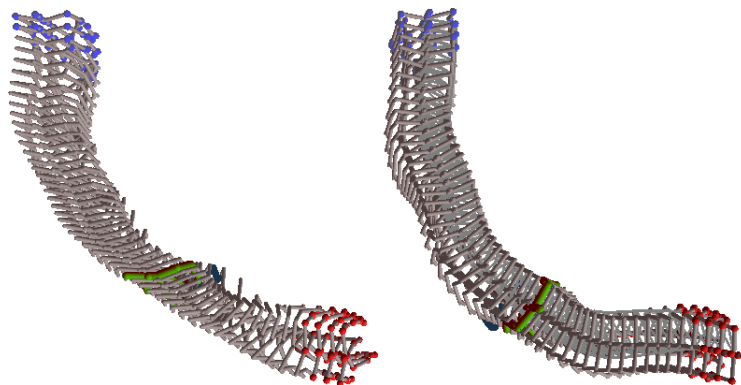


FIGURE 7.5 – Exemple d'édition de graphe sur un mouvement de marche avec les contraintes de rotation (à droite) et sans contraintes de rotation (à gauche). En vert, le squelette de l'animation générée, en gris le graphe, et en bleu et rouge les points de contrôle.

## 7.4 Application à l'édition de mouvement

Notre objectif ici est de fournir un système flexible et complet, facile à utiliser, tout en étant capable de répondre aux questions soulevées par l'édition ou l'adaptation de mouvement. Notre système est interactif, et par ailleurs fonctionne en temps réel. Le seul processus long en temps de calcul est le pré-calcul de la décomposition de Cholesky, effectuée hors ligne. Dans les différentes expériences, l'utilisateur peut choisir et modifier un ou plusieurs points de contrôle du graphe. De plus hormis des distances induites par les segments du squelette, nous avons ajouté la possibilité pour l'utilisateur de spécifier des contraintes de distance entre des articulations non adjacentes ou avec l'environnement.

Le code est écrit en C++ et notre application a été testée sur un ordinateur muni d'un processeur de 2.7 Ghz pour 3Go de RAM.

### 7.4.1 Édition de mouvement

La première application aborde le problème général de l'édition de mouvement. En fixant un certain nombre de sommets du graphe et en déplaçant d'autres sommets, notre système nous permet d'éditer de manière significative le mouvement tout en conservant ses caractéristiques spatio-temporelles. Notre méthode est adaptée à tout type de mouvement impliquant des

rotations de grande amplitude, comme la marche, la course, la boxe, etc., et plusieurs types d'édition tels que les translations et les rotations. Les figures 7.4 et 7.5 ainsi que la vidéo jointe illustrent bien ses performances.

Dans l'expérience illustrée par la figure 7.7, nous avons édité un mouvement (fourni par la base de données CMU) riche en information, il comprend les mouvements dynamiques du corps et des pieds ainsi que des accélérations et décélérations. L'objectif de cette démonstration est d'éditer spatialement la position du pied droit lors de contacts.

La figure 7.6 montre la convergence de notre méthode, à la fois pour l'énergie liée aux contraintes de distance et pour l'énergie globale du système, à l'issue de 10 itérations de minimisation. Les résultats montrent que l'énergie globale  $E$  du système diminue, tout en appliquant la minimisation sur les contraintes liées aux longueurs des segments, le tout en quelques itérations. Cela illustre le succès de notre méthode de minimisation alternée.

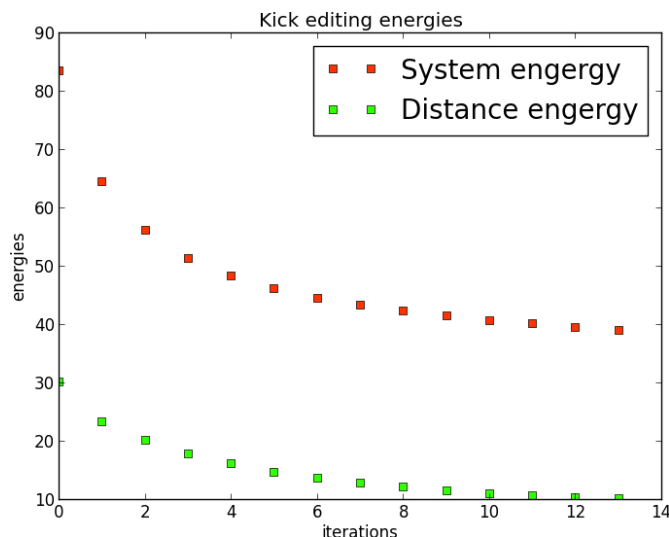


FIGURE 7.6 – Convergence du système et énergie liée aux contraintes de distance pour l'exemple 7.7.

Visuellement notre système répond aux objectifs recherchés : le pied droit est en mesure d'atteindre les cibles tandis que le reste de l'animation est préservé. La courbe extraite illustrée par la figure 7.7 montre que l'accélération (révélant la dynamique du mouvement) est correctement respectée. La courbe de la séquence éditée a en effet les mêmes caractéristiques que le mouvement de référence et celui "vérité terrain". Dans la vidéo, nous reprenons le même exemple en le comparant avec les résultats obtenus par la méthode proposée par [43]. Notre méthode limite le glissement des pieds et s'avère plus cohérente pour une préservation de la dynamique du mouvement.

Plusieurs expériences d'édition de mouvement sont comparées. Les résultats concernant les temps de calcul ou les erreurs de contraintes sont présentés dans le tableau 7.1. Le temps de calcul d'une posture correspond : (i) au calcul du vecteur d'inconnues par la décomposition de

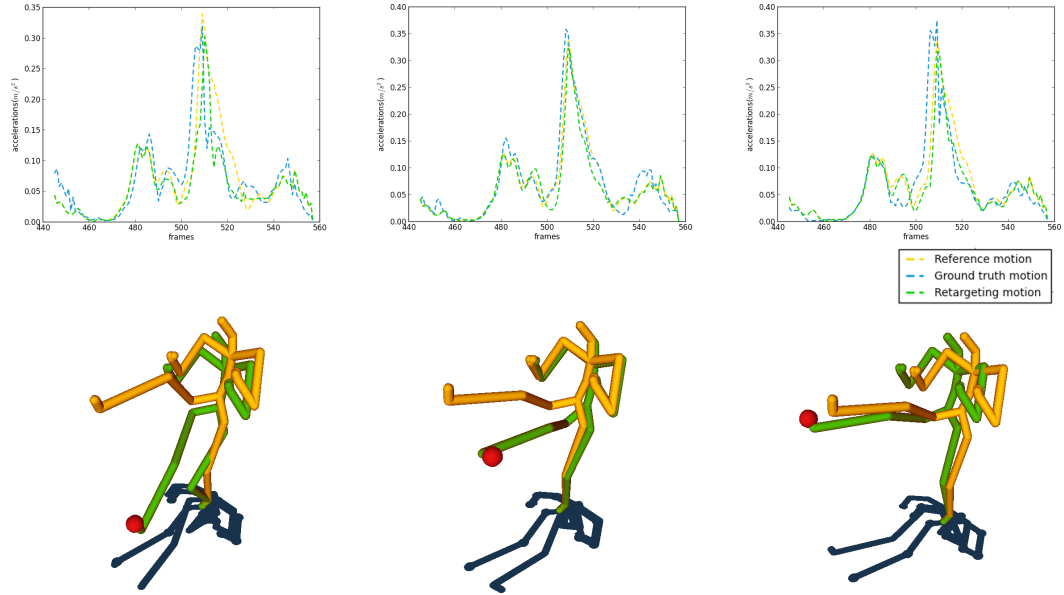


FIGURE 7.7 – Comparaison de 3 cas de cible différentes à atteindre avec le pied droit avec le mouvement de référence (jaune), le mouvement généré (vert) et le mouvement vérité terrain (bleu).

Cholesky ainsi qu'à (ii) la complexité du sous problème dédié aux contraintes de distance et le nombre d'itérations associé. Le pré-conditionnement fourni par la posture précédente, associé à la préservation des relations spatiales par l'opérateur Laplacien, permettent une convergence en peu d'itérations de la méthode. Finalement notre système est interactif pour des séquences représentées par des graphes de 3000 sommets ou moins. Concernant les poids associés aux contraintes, nous avons utilisé les coefficients  $w_u = 1$  et  $w_d = 2$ . Ces poids ont été choisis afin de préserver, en priorité, les longueurs des segments des squelettes. Nous définissons par ailleurs l'erreur de distance relative  $EDR$  correspondant au bon respect des longueurs des segments par :

$$RDE = \frac{\sum_{e(i,j) \in \mathbf{E}_S} \frac{\|v_i^l - v_j^l\|}{d_{ij}}}{|\mathbf{E}_S|} \quad (7.5)$$

avec  $v_i^l$  et  $v_j^l$  étant les sommets à déterminer, à partir des arêtes  $e(i, j)$  et les longueurs originales  $d_{ij}$ . Nous observons que les résultats sont satisfaisants puisque l'erreur de  $RDE$  après 10 itérations est quasi nulle.

Mouvement	nombre de postures	nombre de sommets	Temps pour construire le système de Cholesky (sec.)	Temps pour adapter le système	RDE de la première posture	EDR après 10 postures
Fig 7.4	254	2910	12.48	0.15	0.031	0
Fig 7.5	254	2910	12.06	0.16	0.01	0
Fig 7.7(a)	115	1323	3.2	0.064	0.031	0
Fig 7.7(b)	115	1323	3.17	0.068	0.04	0
Fig 7.7(c)	115	1323	3.2	0.063	0.01	0

TABLE 7.1 – Tableau des résultats pour différents mouvements.



FIGURE 7.8 – Exemple de trois cas d'adaptation morphologique de mouvement. Le squelette jaune est adapté pour 3 morphologies différentes.

## 7.4.2 Adaptation de mouvement

Notre méthode est également appropriée pour adapter des mouvements (*retargeting*). L'objectif consiste à adapter le mouvement d'un personnage vers un autre dont la morphologie est différente. Comme proposé par Gleicher [60], nous nous concentrons ici sur l'adaptation morphologique de personnages articulés ayant une structure identique mais avec des longueurs de segment différentes. En effet, notre méthode nous permet facilement d'ajouter dans le graphe de référence de nouvelles contraintes de distance extraites d'un autre squelette. De plus avant de lancer la minimisation, l'utilisateur peut identifier certaines caractéristiques spécifiques du mouvement comme des contraintes qui doivent être maintenues pendant l'édition : les positions fixes de certains sommets ou encore l'ajout de nouvelles contraintes de distance entre certaines articulations. Au cours de la minimisation, l'opérateur Laplacien préserve les propriétés spatio-temporelles du mouvement. Dans cette section, nous démontrons la capacité de notre approche à adapter un mouvement de référence issu de la capture de mouvement sur l'animation de



FIGURE 7.9 – Application à l'adaptation de mouvement couplée à l'interaction entre personnages. En jaune les squelettes de référence, en verts les squelettes adaptés.

plusieurs maillages avec des morphologies différentes.

Pour tester notre méthode, nous disposons des squelettes de référence et celui adapté au maillage cible (construit manuellement). Nous re-dimensionnons alors les deux squelettes au même facteur d'échelle et sélectionnons certains sommets comme des contraintes de position, avant d'effectuer la minimisation. Cette étape est effectuée afin de fixer l'animation dans l'espace 3D (rappelons que les coordonnées Laplaciennes sont invariantes aux translations) ou encore pour préserver certains contacts. Les résultats sont illustrés par la figure 7.8. Nous pouvons constater visuellement que notre système est robuste à de fortes différences morphologiques, et cette interprétation visuelle est confirmée par la mesure *EDR* (différence de taux entre les longueurs des segments). Ici aussi, l'opérateur Laplacien associé à des contraintes de distance maintient le caractère dynamique du mouvement. Enfin, l'application de l'adaptation du mouvement à la déformation de maillage montre que les résultats sont cohérents.

Comme illustré par la figure 7.9 (et dans la vidéo qui accompagne ce document), nous avons également appliqué la méthode avec succès à de l'adaptation morphologique pour de l'interaction entre personnages. Dans cette expérience, les deux personnages se serrent la main, ainsi nous avons ajouté des contraintes de distance spécifiques entre les mains pour préserver l'interaction. Enfin, cette expérience montre la possibilité de combiner l'édition de mouvement avec de l'adaptation de mouvement.

## 7.5 Disussion et perspectives

Dans ce chapitre, nous avons proposé une nouvelle représentation du mouvement par le *graphe 3D+t*, encodant dans une même structure l'information géométrique des squelettes au cours du temps. Avec cette nouvelle représentation du mouvement, nous avons présenté une méthode d'édition de mouvement qui est en mesure de préserver les caractéristiques essentielles du mouvement et en particulier sa **dynamique** ainsi que ses propriétés **spatio-temporelles**. Notre méthode interactive est temps réel et peut gérer une large gamme de mouvements, avec



les avantages d'un contrôle simple et intuitif pour l'utilisateur.

Toutefois, la préservation des accélérations et décélérations n'assure pas forcément la préservation de l'ensemble des caractéristiques de la dynamique telles que celles liées aux contraintes physiques du mouvement. Parmi les perspectives possibles de notre étude, nous aimerions considérer la préservation d'autres propriétés, par exemple le *Zero Moment Point* ou la projection du centre de gravité. Cette démarche serait réalisable par une paramétrisation des poids du graphe contrôlé par un modèle physique. Par ailleurs, l'opérateur Laplacien 3D+t qui encode l'ensemble de l'information du mouvement semble adapté pour la construction d'espaces de faibles dimensions, ce qui pourrait présenter un intérêt dans le cadre de la recherche de mouvement, ou dans des opérations d'adaptation de mouvement plus complexes, impliquant des interactions avec l'environnement ou des groupes de personnages. Enfin d'un point de vue géométrique, il serait également intéressant d'intégrer dans la boucle de minimisation alternée des contraintes liées aux limites angulaires, ce qui assurerait des résultats plus cohérents. Ce point est développé dans un autre contexte dans le chapitre 9.

L'inconvénient majeur de notre système est de produire à l'issue de l'édition les positions des articulations des squelettes (et non les rotations). Ceci dit, comme indiqué dans l'étude [78], il est tout à fait possible de retrouver les rotations des articulations par de la cinématique inverse, ou encore en utilisant des techniques récentes [15, 26] pour déformer le maillage guidé uniquement par les positions. Nous présenterons dans le chapitre 9 une technique de déformation de maillage qui répond à ce problème.

# 8

## Application à la reconstruction de trajectoires de marqueurs

### Sommaire

---

<b>8.1 Introduction</b>	<b>99</b>
<b>8.2 Approches connexes</b>	<b>100</b>
8.2.1 Méthodes dédiées à la reconstruction de mouvement	100
8.2.2 Utilisation de l'opérateur Laplacien pour de l'interpolation de données	102
<b>8.3 Constats sur le mouvement capturé</b>	<b>102</b>
<b>8.4 Reconstruction du mouvement par la préservation des relations spatiales</b>	<b>105</b>
<b>8.5 Résultats</b>	<b>106</b>
<b>8.6 Conclusion et perspectives</b>	<b>109</b>

---

### 8.1 Introduction

Nous avons présenté dans le chapitre précédent une première application de notre méthode d'édition de graphe destinée à l'édition de mouvement de squelette. Rappelons que notre méthode permet la préservation des relations spatiales à travers un couplage de l'information différentielle du graphe et des contraintes de distance associées à certaines arêtes. Nous présentons dans cette section une seconde application de notre méthode appliquée à la reconstruction de trajectoires de marqueurs.

La capture de mouvement réalisée par un système classique tel que Vicon [175] ou Qualisys [1] fournit les trajectoires des positions en 3D des marqueurs dans l'espace et dans le temps. Toutefois, à l'issue de l'acquisition il existe un certain nombre de trajectoires manquantes dues aux occultations lors de la capture. Cela arrive par exemple lors de la capture des mains où il existe de nombreux contacts et occultations d'une main par l'autre (certains marqueurs sont invisibles pour les caméras du système). La reconstruction complète et précise des trajectoires des mains s'avère pourtant importante pour retranscrire précisément les postures des mains qui peuvent avoir une signification particulière.

Reconstituer ces données manquantes est un problème difficile. Les durées de ces occultations peuvent être plus ou moins longues : quelques images à plusieurs secondes et les trajectoires sont mathématiquement aléatoires. Par ailleurs, en plus des trajectoires dans l'espace, il est impératif de préserver la dynamique temporelle du mouvement qui caractérise le style ou une certaine forme d'expressivité des gestes. D'autre part, les mouvements capturés étant souvent destinés à l'animation de personnages articulés, il est important de préserver la cohérence spatiale des mouvements reconstruits, notamment en s'assurant que certaines distances entre les marqueurs sont conservées durant la reconstruction.

Lors de différentes expérimentations, nous avons constaté que l'animation d'un marqueur était souvent liée à celle de son voisinage. Dans cette section du manuscrit, nous proposons une alternative aux méthodes classiques pour reconstituer les trajectoires des marqueurs occultés en prenant en considération l'information de voisinage d'un marqueur sous la forme de contraintes dures et souples. Les **contraintes souples** correspondent aux informations spatiales et temporelles du mouvement, elles sont à préserver au mieux alors que les **contraintes dures** correspondent au bon respect de certaines distances implicites induites par le squelette sous-jacent.

Notre étude conduit à trois contributions majeurs : (i) nous proposons une nouvelle représentation du mouvement par la construction d'un graphe encodant les relations spatiales entre les marqueurs via l'opérateur Laplacien. Cette représentation permet d'établir une interdépendance entre les marqueurs et *in fine* de préserver certaines relations spatiales durant la reconstruction. (ii) notre méthode permet une préservation des distances correspondant aux longueurs des segments du squelette sous-jacent. (iii) pour reconstruire le mouvement, nous proposons une technique de reconstruction originale basée sur un algorithme double-passe : dans le sens du temps puis dans le sens inverse, chacune des passes reconstruisant les configurations du nuage de marqueurs par la méthode de minimisation alternée développée dans le chapitre 6.

Avant de d'écrire notre méthode, nous présentons ci-après quelques approches connexes. Nous appliquons ensuite notre méthode à la reconstruction de mouvements capturés.

## 8.2 Approches connexes

### 8.2.1 Méthodes dédiées à la reconstruction de mouvement

Il existe un grand nombre de méthodes dédiées à la reconstruction de trajectoires incomplètes issues de la capture de mouvement. Li et co-auteurs [106] les ont classifiées en plusieurs catégories :

**Interpolation linéaire et par spline** Souvent fournies avec les logiciels connus (3DSMax, Maya, Qualisys), ces méthodes sont davantage orientées vers le traitement d'occultations de

courtes durées. Elles permettent une reconstruction couplant l'information spatiale et temporelle mais traitent les marqueurs indépendamment les uns des autres.

**Méthodes basées squelette** L'utilisation des positions des marqueurs pour retrouver l'animation du squelette est un processus très répandu. Plusieurs études [32, 73, 97] utilisent des méthodes géométriques pour établir la démarche inverse en utilisant l'information de structure du squelette pour reconstruire les trajectoires des marqueurs perdus. [197] utilise un système physique guidé par l'animation des marqueurs pour générer l'animation du squelette. Le système contraint par un ensemble de forces permet de retrouver les trajectoires des marqueurs manquants. En utilisant une structure de squelette, ces méthodes permettent (i) d'établir à travers les liens entre les articulations une interdépendance entre les marqueurs (ii) de préserver les distances entre certains marqueurs puisque l'espace des rotations préserve la longueur des segments. Toutefois, le passage de l'espace des positions à celui des angles est approximatif et la structure en arbre du squelette limite l'interdépendance entre les marqueurs. De plus, ces méthodes traitant l'information indépendamment d'une posture à une autre intègrent difficilement les aspects temporels du mouvement.

**Réduction de dimension** Plusieurs approches comme [111, 162] proposent de projeter linéairement les positions des marqueurs sur les axes principaux pour reconstruire les morceaux de trajectoires occultés. Par exemple, Park et Hodgins [136] utilisent une analyse en composantes principales (ACP) pour estimer les trajectoires des marqueurs manquants dans le cadre de la capture de déformation de peau. Il existe d'autres études basées sur des modèles non linéaires. Taylor et co-auteurs [169] utilisent par exemple des réseaux de neurones (Machine de Boltzmann Restreinte (CRBM)) appliqués à la génération de mouvement humain. Leur approche qui utilise des rotations en entrée permet à partir d'un apprentissage sur les *frames* antérieures de retrouver un jeu de paramètres caractérisant la posture à générer.

**Utilisation de bases de données existantes** D'autres méthodes comme [80] proposent d'utiliser une base de données existante dont les mouvements sont caractérisés par un ensemble de propriétés faisant le lien entre les mouvements incomplets et ceux de la base. Chai et Hodgins [34] utilisent des ensembles de marqueurs de faible dimension comme signal de contrôle pour rechercher un mouvement dans la base de données. Ces techniques permettent de bons résultats mais nécessitent toutefois un grand nombre de données de bonne qualité, traitées souvent manuellement. De plus elles ne permettent pas la génération de nouveaux mouvements.

**Systèmes dynamiques** Que ce soit dans l'espace des positions ou des rotations, de nombreuses méthodes reposent sur des systèmes dynamiques comme par exemple celles utilisant les filtres de Kalman [6, 50, 150]. La difficulté de ces systèmes est d'en trouver une bonne paramétrisation, ainsi plusieurs études [6, 150] proposent des méthodes pour les apprendre.

Une autre technique consiste à utiliser des processus Gaussiens [177]. [43] propose un modèle stochastique appliqué à l'espace des rotations pour générer aléatoirement les trajectoires manquantes à partir d'informations apprises sur le mouvement. Plus récemment Li et co-auteurs [106] proposent à travers leur système *BoLerO*, une approche reposant sur un modèle stochastique dont les objectifs sont particulièrement proches des nôtres, leurs travaux sont basés sur un modèle stochastique qui tient compte des corrélations entre les marqueurs et des dynamiques à travers le temps. De notre côté, nous mettons en évidence ces corrélations à travers l'existence de relations spatiales entre les marqueurs. L'autre point commun concerne l'inclusion de contraintes de distance liées à la longueur des segments.

### 8.2.2 Utilisation de l'opérateur Laplacien pour de l'interpolation de données

Le principe d'interpolation entre deux graphes  $G_a$  et  $G_b$  de même connectivité est de déterminer  $n$  configurations spatiales transformant  $G_a$  vers  $G_b$ . Plusieurs méthodes utilisant l'opérateur Laplacien pour de l'interpolation de données ont déjà été proposées. Par exemple Alexa [2] et Sorkine [156] utilisent les coordonnées différentielles pour interpoler entre deux maillages. D'un autre côté, Takahashi [167] propose dans le cadre de simulation de foule une méthode d'interpolation entre deux configurations de graphe de même connectivité basée sur l'étude spectrale des matrices Laplaciennes des deux graphes. Ces méthodes obtiennent respectivement dans leurs domaines de bons résultats mais ne sont pas applicables dans notre contexte. En effet, ces deux méthodes nécessitent les graphes de référence  $G_a$  et  $G_b$  complets (même nombre d'éléments dans les graphes) et calculent l'intégralité du graphe entre  $G_a$  et  $G_b$ . De notre côté, nous ne sommes pas sûr de disposer de l'intégralité des graphes  $G_a$  et  $G_b$ , de plus nous disposons pour chaque graphe allant de  $G_a$  vers  $G_b$  de configurations partiellement connues.

Dans la section suivante nous proposons de faire plusieurs constats sur le mouvement capturé motivant notre démarche avant d'appliquer notre méthode développée dans le chapitre 6 à ce problème bien spécifique de reconstruction de graphe.

## 8.3 Constats sur le mouvement capturé

Dans cette section nous présentons quelques constats sur différents mouvements et en particulier ceux des mains afin de justifier notre démarche. Nous nous sommes focalisés sur les mains qui sont des structures multi-articulées complexes qui permettent de bien illustrer notre problématique. Cette dernière pourrait également être appliquée à la reconstruction du reste du corps ainsi qu'à des nuages de points issus de l'animation faciale.

**Représentation du mouvement de référence** A l'issue de la capture de mouvement, nous obtenons un vecteur de positions évoluant dans le temps. Après une phase d'étiquetage permettant l'identification des marqueurs, nous obtenons des trajectoires discrétisées en  $x$ ,  $y$ , et



FIGURE 8.1 – Reconstruction de la trajectoire en X (rouge), Y (vert) et Z (bleu) du marqueur du menton avant (en haut) et après (en bas) l’utilisation de notre méthode d’interpolation.

$z$  pour chaque marqueur. Mathématiquement, les positions des marqueurs forment une série temporelle multi-dimensionnelle de dimension  $3n \times m$  avec  $n$  le nombre de marqueurs et  $m$  le nombre de *frames* capturées. Ces trajectoires sont très souvent incomplètes, ce qui est dû aux occultations lors de l’opération de capture. La figure 8.1 (haut) illustre bien le problème d’une trajectoire possédant 5 occultations allant de 2 à 300 *frames*.

**Remarques** Nous avons constaté qu’il existait pour beaucoup de marqueurs une corrélation plus ou moins forte entre son déplacement et ceux de ses voisins. Par exemple pour la capture faciale, les marqueurs positionnés sur les zones rigides comme le front, le menton ou encore les joues ont un déplacement semblable. Dans tous les cas, il existe un lien entre un marqueur et n’importe quel autre marqueur du visage (à travers leur transformation commune liée à la rotation de la tête). Concernant les mouvements de la main, de nombreuses références ont mis en évidence des corrélations **spatiales** entre les articulations d’un même doigt ou encore entre les mouvements des doigts de la main (*e.g* [32,37,104,137]). La figure 8.2 qui représente les dynamiques (accélérations) des articulations de l’annulaire et du majeur au cours du temps montre bien qu’il existe une corrélation **temporelle** d’une articulation avec ses voisines. La table 8.1

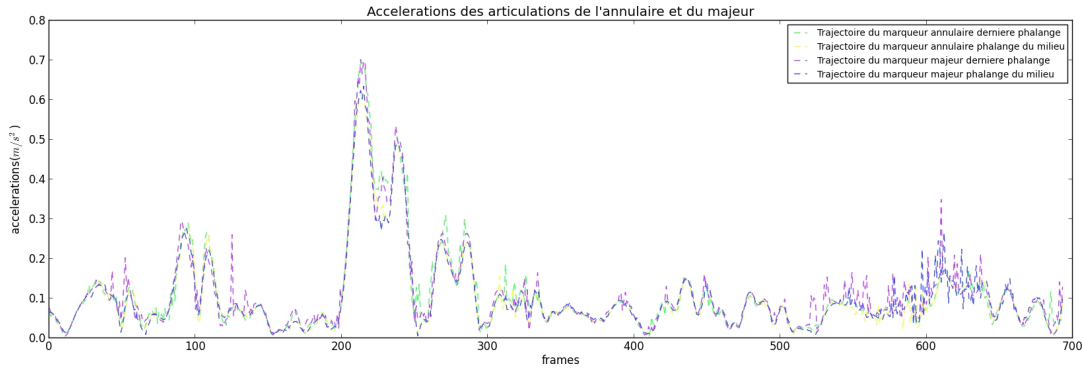


FIGURE 8.2 – Accélérations des articulations des phalanges du milieu et extrémité de l’annulaire et du majeur.

donne l’erreur relative<sup>1</sup> de l’accélération entre l’extrémité de l’annulaire avec les autres articulations du majeur et de l’annulaire. Elle montre que les dynamiques sont proches ce qui confirme notre premier constat. Par ailleurs, nous avons observé que plusieurs distances entre certains marqueurs correspondant aux articulations adjacentes d’un squelette sous-jacent sont maintenues au cours de l’animation. Il s’agit en particulier des paires de marqueurs disposés au niveau des articulations adjacentes. Comme il est mentionné par Li [106], cette caractéristique des mouvements articulés est une information essentielle parmi les propriétés du mouvement.

	milieu annulaire	extrémité majeur	milieu majeur
Erreur relative $E_{rel}$	0.082%	0.48%	0.497%

TABLE 8.1 – Erreurs relatives (note de bas de page 1) des accélérations entre l’extrémité de l’annulaire et les articulations du milieu et de l’extrémité de l’annulaire et du majeur.

**Hypothèse** A partir de ces différentes remarques, nous faisons l’hypothèse que les corrélations entre les différents marqueurs peuvent être encodées sous la forme de relations spatiales. Dans la section suivante, nous proposons une nouvelle représentation du mouvement des marqueurs en ajoutant à la représentation actuelle les informations de relations spatiales couplées aux contraintes de distance. Cette démarche qui consiste à passer par une structure qui lie les éléments caractérisant le mouvement (marqueurs, articulations) entre eux est similaire aux méthodes basées squelette (*e.g* [73, 97]) puisque : elle permet de donner une information d’interdépendance entre les marqueurs. Toutefois en utilisant l’espace des positions des marqueurs, notre approche est plus proche des travaux de Li et de ses collaborateurs [106].

1. Nous définissons l’erreur relative entre les deux courbes d’accélération  $a_1$  et  $a_2$  par la fonction :

$$E_{rel}(a_1, a_2) = \frac{1}{m} \sum_{i=0}^m \left( \frac{a_1(i) - a_2(i)}{a_2(i)} \right)^2, \quad (8.1)$$

où  $m$  est le nombre de frames.

## 8.4 Reconstruction du mouvement par la préservation des relations spatiales

Afin de reconstituer les configurations des nuages de points pour chaque posture dans leur intégralité, nous utilisons l'opérateur Laplacien discret. Comme nous l'avons vu précédemment, l'opérateur Laplacien permet d'exprimer un point vis-à-vis de ses voisins dans l'espace cartésien par une information différentielle associée à chaque point. Plus précisément, notre démarche se focalise sur la reconstruction de trajectoires occultées en cherchant à préserver les relations spatiales entre un marqueur et ses voisins (**contraintes souples**), couplé à des contraintes de distance entre certains marqueurs (**contraintes dures**). Ces contraintes de distance peuvent par exemple être définies pour les paires de marqueurs situés au niveau d'articulations adjacentes. Bien que ces contraintes sont approximatives (il est plus logique de considérer les segments du squelette allant d'un centre articulaire vers un autre centre articulaire adjacent), elles permettent d'obtenir une meilleure reconstruction du mouvement (cette caractéristique est discutée dans les travaux de Li et co-auteurs [106]).

**Représentation du mouvement** Les données dont nous disposons sont caractérisées par un nuage de points évoluant au cours du temps. Or l'utilisation de l'opérateur Laplacien requiert une information de connectivité entre les sommets. Ainsi nous représentons le mouvement par un vecteur de graphe. A chaque posture  $i$  est associée un graphe  $G_i = (\mathbf{V}_i, \mathbf{E}_i)$  dans lequel  $\mathbf{V}_i$  représente les positions des marqueurs à la posture  $i$  et  $\mathbf{E}_i$  les liens entre les marqueurs les plus corrélés. Nous laissons le soin à l'utilisateur de constituer l'information de connectivité du graphe qui lui semble la plus cohérente. Cette étape permet de donner plus ou moins d'importance entre un point et ses voisins. De plus nous associons aux arêtes dont les longueurs doivent rester fixes une contrainte de distance. Un exemple de la création d'un tel graphe (dans le cas de mains) est donné par la figure 8.3. Les liens gris représentent les arêtes du graphe alors qu'en pointillé rouge sont représentées les contraintes de distance. La connectivité du graphe ainsi que les contraintes de distance ne sont données qu'une seule fois et appliquées à l'ensemble du mouvement.

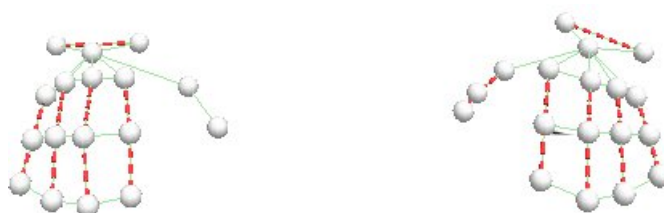


FIGURE 8.3 – Illustration de la construction du graphe des voisins et contraintes de distance.

Dans le paragraphe suivant, nous montrons comment étendre notre méthode d'édition de graphe donnée dans la section 6 pour reconstruire les trajectoires des marqueurs manquantes.



**Préservation des relations spatiales** Pour reconstituer au mieux le mouvement, nous étendons la méthode précédemment introduite en préservant les relations spatiales ainsi que la dynamique du mouvement entre le marqueur manquant et ses voisins.

Pour reconstruire une posture du mouvement, l'approche consiste à d'utiliser la méthode d'édition de graphe en prenant les configurations spatiales des postures qui lui sont temporellement voisines (posture d'avant et d'après). Par ailleurs, nous utilisons des poids uniformes pour paramétrer l'opérateur Laplacien : nous donnons ainsi autant d'importance aux voisins d'un même sommet. Cette option est sujette à amélioration et discutée dans les perspectives de ce chapitre.

Plus précisément, nous utilisons la bonne configuration spatiale d'une image  $i - 1$  (l'ensemble des marqueurs sont présents) pour corriger la posture  $i$  suivante. Ainsi nous cherchons à corriger la posture  $i$  en préservant l'expression Laplacienne du graphe à la posture  $i - 1$  à laquelle nous ajoutons les contraintes de position des marqueurs non occultés en  $i$ . De plus nous nous assurons que les contraintes de distance données par l'utilisateur sont bien respectées. Le processus itératif permettant de reconstruire les trajectoires des marqueurs suivant le principe d'édition de graphe, est décrit par l'algorithme 2. La seconde passe de l'algorithme permet de reproduire ce schéma itératif dans un sens inverse, *i.e* en utilisant la posture  $i + 1$  pour corriger la posture  $i$ . Enfin nous fusionnons les résultats des deux trajectoires des deux passes par une interpolation linéaire. D'un point de vue temporel, ce sont les contraintes de position données par les marqueurs de la posture  $i$  non occultés qui permettent la préservation de la dynamique.

Notons que notre méthode est analogue à l'algorithme *forward-backward* [139]. En effet, cet algorithme qui calcule la probabilité d'une séquence observée dans le contexte des modèles de Markov cachés se décompose en trois phases : calcul progressif (1) et rétrogressif (2) des probabilités, puis une combinaison (3) des deux ensembles de probabilités des états cachés.

## 8.5 Résultats

Afin de tester et d'utiliser notre méthode de manière interactive, nous avons développé en C++ un logiciel sous QT (voir les fichiers attachés à ce présent document) dédié à la reconstruction de données. Pour acquérir nos données nous avons utilisé le système Qualisys [1] constitué d'un ensemble de 7 caméras associé au logiciel *QTM* permettant d'obtenir et d'étiqueter les trajectoires. L'ensemble des expérimentations ont été capturées à la fréquence de 150 Hz.

A partir d'un ensemble de trois séquences de capture de mouvement du haut du corps (mains, bras et visage) d'environ 4500 frames, nous avons observé que la perte de signal sur une séquence est d'environ 5% avec comme maximum une perte de 11% pour le marqueur le plus occulté. Le nombre moyen de postures pour une occultation est de 90 et en moyenne chaque marqueur subit 4 occultations.

A titre indicatif, le temps de calcul de notre méthode pour une séquence de 4447 images avec 52 marqueurs et 4% de perte de données est d'environ 30 secondes. L'application pourrait largement être optimisée en isolant les parties de graphe où les occultations apparaissent

**Données:**  
**V** : vecteur des positions des marqueurs  
**V'** : vecteur des positions des marqueurs interpolés  
**G** : le graphe associé

- 1 //Première passe ;
- 2 **pour** *l'ensemble des postures du mouvement i : 1 jusqu'à n faire*
- 3     
$$\mathbf{V}_i^1 = \operatorname{argmin}_{\mathbf{V}_i} \begin{pmatrix} \mathbf{L}_{i-1} \\ \mathbf{U}_i \end{pmatrix} \mathbf{V} = \begin{pmatrix} \Delta_{i-1} \\ \mathbf{u}_i \end{pmatrix}$$
- 4     **tant que pas de convergence faire**
- 5         calcul de  $\hat{\Gamma}$  à partir de  $\operatorname{argmin}_{\mathbf{V}} E_d(\mathbf{V}, G)$  (équation 6.4), préconditionné avec  $\mathbf{V}_i^1$  ;
- 6         
$$\mathbf{V}_i^1 = \operatorname{argmin}_{\mathbf{V}_i} \begin{pmatrix} \mathbf{L}_{i-1} \\ w_u \mathbf{U}_i \\ w_d \mathbf{D} \end{pmatrix} \mathbf{V} = \begin{pmatrix} \Delta_{i-1} \\ w_u \mathbf{u}_i \\ w_d \hat{\Gamma} \end{pmatrix}$$
- 7 //Seconde passe ;
- 8 **pour** *l'ensemble des images du mouvement i : n - 1 jusqu'à 0 faire*
- 9     
$$\mathbf{V}_i^2 = \operatorname{argmin}_{\mathbf{V}_i} \begin{pmatrix} \mathbf{L}_{i+1} \\ \mathbf{U}_i \end{pmatrix} \mathbf{V} = \begin{pmatrix} \Delta_{i+1} \\ \mathbf{u}_i \end{pmatrix}$$
- 10     **tant que pas de convergence faire**
- 11         calcul de  $\hat{\Gamma}$  à partir de  $\operatorname{argmin}_{\mathbf{V}} E_d(\mathbf{V}, G)$  (équation 6.4), préconditionné avec  $\mathbf{V}_i^2$  ;
- 12         
$$\mathbf{V}_i^2 = \operatorname{argmin}_{\mathbf{V}_i} \begin{pmatrix} \mathbf{L}_{i-1} \\ w_u \mathbf{U}_i \\ w_d \mathbf{D} \end{pmatrix} \mathbf{V} = \begin{pmatrix} \Delta_{i-1} \\ w_u \mathbf{u}_i \\ w_d \hat{\Gamma} \end{pmatrix}$$
- 13 //Interpolation finale
- 14 **pour** *l'ensemble des occultations faire*
- 15     //k : l'index de l'image dans l'intervalle de l'occultation
- 16     //p : nombre d'images manquantes pour une occultation
- 17     **pour** *de l'image k : 0 jusqu'à p faire*
- 18         
$$\mathbf{V}_i' = (1 - (k/p)) * \mathbf{V}_i^1 + (k/p) * \mathbf{V}_i^2$$

**Algorithm 2:** PROCESSUS DE RECONSTRUCTION DE TRAJECTOIRES DE MARQUEURS OCCULTÉS

mais ce traitement étant actuellement dédié à un processus off-line, nous n'avons pas estimé intéressant d'effectuer cette optimisation. La figure 8.1 présente les différents résultats avant et après interpolation. Nous pouvons observer dans un premier temps que la courbe originale est très fortement non linéaire et le résultat de l'interpolation l'est également. D'un point de vue visuel, nous pouvons voir dans la figure 8.1 et dans la vidéo associée à ce document que les résultats paraissent réalistes.

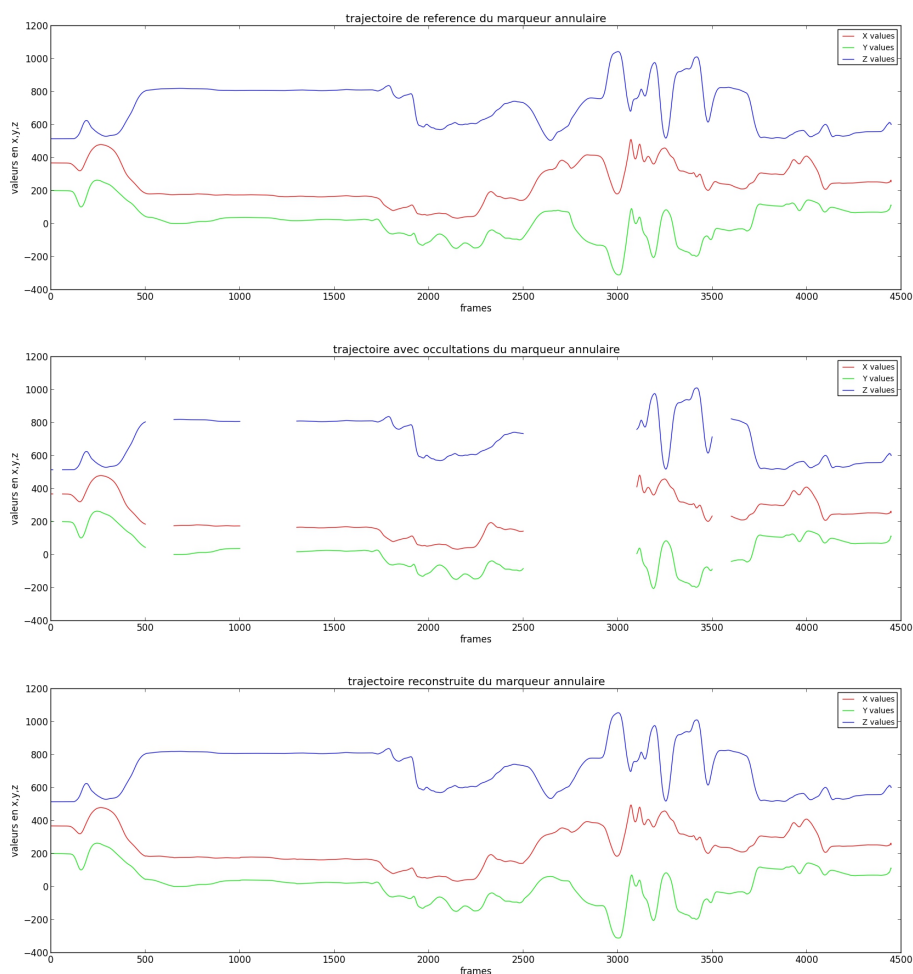


FIGURE 8.4 – Reconstruction de la trajectoire en X (rouge), Y (vert) et Z (bleu) du marqueur du menton avant (en haut) et après (en bas) l'utilisation de notre méthode d'interpolation.

Dans un second temps nous avons testé notre méthode sur le mouvement de la dernière articulation de l'annulaire en enlevant volontairement des morceaux de trajectoire sur des courbes originales afin de comparer le résultat de l'interpolation. Dans l'expérience illustrée par la figure 8.4, nous avons introduit artificiellement 5 occultations : des frames 10 à 60, 500 à 650,

1000 à 1300, 2500 à 3100 et de 3500 à 3600, ce qui représente une perte de l'information sur l'ensemble du mouvement du marqueur de 27%. Après reconstruction, notre méthode accomplit la tâche avec succès puisqu'elle nous donne une erreur relative de 0.02%.

Dans les deux cas d'études précédemment décrits, les contraintes de distance sont bien préservées en peu d'itérations, le grand nombre de contraintes de position donnant un très bon pré-conditionnement à la descente de gradient.

## 8.6 Conclusion et perspectives

Dans cette section nous avons présenté une nouvelle application de notre méthode couplant l'utilisation de l'opérateur Laplacien associé à des contraintes de distance pour de la reconstruction de trajectoire dans l'espace des positions. A travers une représentation du nuage de marqueurs via un graphe encodant l'information spatiale pour chaque *frame*, l'opérateur Laplacien permet de préserver les relations spatiales entre le marqueur occulté et ses voisins ainsi que de prendre en compte implicitement les propriétés temporelles de la trajectoire grâce à la dynamique des marqueurs voisins. Les contraintes de distance permettent également de préserver les bonnes longueurs entre certains marqueurs.

A travers les résultats obtenus, nous avons montré que notre méthode, intuitive à utiliser, permet des reconstructions réalistes malgré de fortes perturbations sur le signal.

Parmi les perspectives, il serait envisageable d'appliquer notre technique à d'autres contextes de données multi-variées. D'un point de vue technique, nous pensons que nous pourrions améliorer sensiblement notre système en paramétrant les poids laplaciens en fonction de l'importance des voisins d'un sommet. Manuellement, cette démarche donnerait plus de travail à l'utilisateur. Une autre possibilité consisterait à extraire les corrélations statistiques dans les données présentes pour paramétrer les poids. Enfin nous pensons qu'il serait intéressant de reformuler le problème appliqué à l'algorithme *forward-backward* [139] où chaque état caché  $i$  serait calculé par une probabilité de réalisation.



# 9

## Contrôle de maillage temps réel guidé par des marqueurs issus de la capture de mouvement

### Sommaire

---

<b>9.1</b>	<b>Introduction</b>	<b>111</b>
<b>9.2</b>	<b>Approches connexes</b>	<b>114</b>
9.2.1	Déformation de maillage	114
9.2.2	Animation de maillage	115
9.2.3	Notre approche	120
<b>9.3</b>	<b>Présentation du système</b>	<b>120</b>
<b>9.4</b>	<b>Déformation du maillage volumique incluant les propriétés du squelette</b>	<b>122</b>
9.4.1	Couplage squelette/maillage via la construction du maillage volumique	122
9.4.2	Déformation de maillage As-Rigid-As-Possible	123
9.4.3	Préservation des propriétés du squelette	127
9.4.4	Projection du squelette dans l'espace des postures admissibles	129
9.4.5	Relâchement des contraintes d'ARAP autour des articulations	133
9.4.6	Déformation finale du maillage de référence	135
9.4.7	Algorithme	136
<b>9.5</b>	<b>Implémentation et résultats</b>	<b>136</b>
9.5.1	Tests basiques	137
9.5.2	Édition de maillage guidée par des sommets définis manuellement	138
9.5.3	Animation de maillage guidée par le squelette	140
9.5.4	Animation de maillage guidée par les marqueurs	142
<b>9.6</b>	<b>Conclusion et perspectives</b>	<b>144</b>

---

### 9.1 Introduction

Déformer et animer interactivement avec réalisme des systèmes articulés est un sujet majeur en informatique graphique. Le processus classique d'animation de personnage est com-

posé de deux phases : dans un premier temps, l'animation du squelette est générée à partir des rotations et positions des articulations dans le temps ; ces données peuvent être directement récupérées à partir des positions des marqueurs provenant de la capture de mouvement. Dans un second temps, la déformation du maillage "attaché" au squelette est guidée par l'animation du squelette.

Comme nous l'avons détaillé dans la section 1.2 du chapitre *introduction*, cette approche peut être approximative pour plusieurs raisons : premièrement l'étape de calcul des centres articulaires à partir des trajectoires des marqueurs est imprécise, c'est en particulier le cas pour l'animation des mains. Par ailleurs, cette phase ne permet pas forcément de reconstituer l'intégralité du squelette, il est alors classique d'avoir recours à l'intervention de designers et l'utilisation de techniques telles que la cinématique inverse pour retrouver les articulations manquantes. Enfin, cette représentation perd l'information de distance entre le centre articulaire et les marqueurs situés sur la peau. Deuxièmement, l'étape de *rigging* classiquement opérée pas des designers amène souvent à des approximations en particulier pour des zones comportant beaucoup d'articulations. De plus elle nécessite une adaptation de la morphologie du squelette avec celle du maillage (*retargeting*), ce qui conduit à déformer les données d'animation originales.

Dans ce chapitre, nous proposons une technique d'animation à partir de données capturées qui ne vise pas à animer le squelette puis à guider le maillage associé ; mais à animer directement le maillage par les trajectoires des marqueurs associées à plusieurs contraintes induites par les propriétés d'un squelette sous-jacent.

Pour ce faire, nous nous appuyons sur plusieurs concepts développés dans la communauté déformation de maillage où l'objectif de la déformation de forme et des méthodes d'éditions ont pour but de préserver localement la forme de l'objet. Par exemple, l'utilisation de l'opérateur Laplacien discret [156, 160] permet des déformations naturelles de bonnes qualités préservant les détails de la géométrie. Si ces méthodes obtiennent de bons résultats dans le domaine de l'édition interactive, elles montrent leurs limites dans l'expression de contraintes rigides, induites par exemple par un squelette interne au maillage. D'autres travaux proposent des méthodes similaires appliquées à l'animation de maillage guidée par le squelette [26, 30], ou par des trajectoires de marqueurs [34, 136, 164], ou [163]. Cependant aucune de ces dernières approches ne simulent la déformation du maillage en incluant implicitement les contraintes induites par le squelette.

**Contributions.** Notre approche établit un couplage implicite entre la déformation du maillage et l'animation du squelette associé. Cela se fait en incorporant dans la structure géométrique une armature associée au squelette, et en prenant en compte **les contraintes de rigidité** ainsi que **les butées articulaires** caractérisant le squelette. A travers cette approche qui ne dissocie pas l'animation du squelette de l'animation du maillage, nous exploitons des méthodes de déformation de maillage qui permettent de conserver des déformations naturelles tout en s'appuyant sur des données de mouvements capturés réalistes.

Pour atteindre ces objectifs, nous proposons une nouvelle méthode d'animation de maillage

interactive en utilisant un maillage de contrôle volumique sur lequel est appliqué la déformation induite par les points de contrôle. Le maillage volumique est créé à partir d'une surface de référence enveloppant la géométrie articulée originale. Les intérêts d'utiliser un tel maillage de contrôle sont multiples : (i) Le maillage est tétraédrique, ainsi il a les propriétés 2-variété impliquant aucune intersection et aucun trou dans le maillage. (ii) Le maillage de contrôle prend en compte les contraintes internes (structure du squelette) ainsi que des contraintes externes (marqueurs ou points de l'environnement). Cette structure volumique est générée en intégrant un maillage *cage* couvrant le maillage de référence initial, les sommets et arêtes qui composent les articulations du squelette, ainsi que les points de contrôle externes tels que les marqueurs. (iii) Le maillage de contrôle est finalement un maillage simplifié encodant l'essentiel de l'information, ce qui améliore le temps de calcul et la convergence de la méthode [21].



FIGURE 9.1 – Illustration du processus à partir de la spécification d'un nombre réduit de marqueurs issus de la capture de mouvement (image de gauche), notre méthode commence par construire le maillage de contrôle volumique (image centrale) en intégrant le maillage surfacique simplifié ainsi que les informations de rigidité (exprimée sous la forme d'un squelette, finalement doté des limites angulaires des articulations) ; ce maillage est utilisé pour animer le maillage de haute résolution (image de droite)

A notre connaissance, notre méthode de déformation de maillage propose plusieurs innovations inédites :

- elle conserve la rigidité globale induite par la structure de squelette : longueurs des segments et contraintes de limites angulaire des articulations.
- Au cours de la déformation, nous préservons les détails géométriques et les rotations locales. Pour se faire, nous proposons une représentation du maillage de contrôle sous une forme volumique, associée à la méthode *as-rigid-as-possible*. Ainsi notre maillage de contrôle est fermé et sans intersections.
- Puisque le maillage est volumique, la méthode préserve le volume de la forme articulée :



l'énergie locale est donc préservée, pas seulement à la surface de la géométrie, mais aussi dans sa structure interne. De plus, notre système améliore la robustesse de la méthode pour de grandes déformations, et empêche les auto-intersections locales.

- La technique est robuste et bien adaptée pour l'animation interactive en temps réel.

Après avoir examiné les travaux antérieurs dans la section 9.2, nous décrivons dans la section 9.3 la méthode pour construire les différentes structures de maillage et expliquer comment la méthode de déformation peut être appliquée à ces structures pour générer des animations réalistes. Nous montrerons ensuite dans la section 9.5 l'intérêt de notre méthode dans plusieurs cas pratiques avec plusieurs jeux de données, avant de conclure et de donner des perspectives de ce travail.

## 9.2 Approches connexes

Générer des animations de personnages réalistes tout en maintenant des déformations plausibles est un sujet de recherche actif en infographie. Dans cette section nous énumérons les travaux connexes à ceux que nous avons développés. Notre contribution étant dédiée à de l'animation de maillage articulé, nous nous focalisons davantage sur les méthodes intégrant des informations de squelette. Nous séparons en deux parties distinctes les méthodes s'adressant uniquement à de la déformation et celles dédiées à de l'animation de maillage. L'ensemble des caractéristiques des approches présentées ci-après est synthétisé dans le tableau 9.1.

### 9.2.1 Déformation de maillage

Pour animer le maillage, notre méthode s'inspire des travaux de déformation de maillage contrôlée par des points de contrôle.

Parmi celles n'utilisant pas d'information de squelette, il existe un grand nombre de méthodes de déformation permettant d'obtenir des postures *statiques*. Ces méthodes souvent basées sur de l'optimisation appliquée à des coordonnées différentielles permettent de préserver les détails de la géométrie. Certaines sont surfaciques [22, 113, 115, 164, 189] et d'autres utilisent des maillages volumiques [35, 82, 155]. D'autres méthodes se basent sur des modèles physiques tels que par exemple la méthode des éléments finis (*e.g* [116]). Comme discuté dans la section 5.3.1, beaucoup d'entre elles utilisent l'opérateur Laplacien pour sa capacité à produire des déformations de qualité [3, 110, 156]. Cependant, la plupart des approches ne permet pas d'appliquer de grandes rotations pendant l'édition. Zhou et col. [194] proposent une méthode de déformation basée sur un modèle tétraédrique issu du maillage surfacique et applique la minimisation Laplacienne directement sur le volume. Cette méthode permet de mieux préserver le volume mais n'est pas optimale pour la préservation des détails de la géométrie lors de rotations de fortes amplitudes. Une autre approche intéressante s'appuie sur la déformation d'une structure simplifiée obtenue dans l'espace réduit du maillage [48]. Cet espace appelé *eigen-skeleton* par les auteurs fournit une structure de contrôle alternative aux méthodes de dé-

formation par *cake*. Le système produit des postures de bonne qualité mais ne préserve pas les propriétés inhérentes au squelette.

**Les méthodes ARAP :** Sorkine et col. [160] (cf. 5.3.1.1 et B) introduisent le paradigme de déformation *as – rigid – as – possible* (ARAP) qui permet de déformer le maillage avec des rotations de grandes amplitudes en préservant la rigidité locale à chaque sommet du maillage. Borosan [21] optimise les temps de calcul en combinant la méthode avec une technique par *cake*. Il utilise ARAP sur la *cake* et déforme le maillage d'origine avec la technique des coordonnées de valeur moyenne (*Mean value coordinates*). Plus récemment Faraj [54] utilise également une technique par *cake* combinée à ARAP pour déformer des maillages volumiques quelconques. Dans son cas, l'utilisation d'un maillage *cake* permet d'avoir un maillage de contrôle différentiable (et permet donc une utilisation d'ARAP), ce qui n'est pas le cas des maillages manipulés. Manson [115] adapte l'algorithme original pour améliorer significativement les temps de calcul en déformant un maillage simplifié par effondrement d'arête et inverse le processus pour reconstruire le maillage de référence. D'un autre côté, Zhang [190] étend ARAP avec la notion de rigidité interne au maillage en intégrant un squelette approximatif. Les résultats donnent des déformations avec une meilleure préservation du volume sans toutefois assurer totalement la rigidité le long des segments ainsi que la préservation de leur longueur. Zhao et col. [192] proposent également une amélioration de la préservation du volume en utilisant un modèle volumique intégrant des cubes placés manuellement aux endroits sensibles à l'intérieur du maillage. La méthode de résolution est similaire à celle d'ARAP pour calculer le composant de rotation à appliquer.

### 9.2.2 Animation de maillage

A l'opposé des méthodes de déformation dédiées au design de postures statiques, l'animation de maillage doit calculer pour chaque pas de temps des postures cohérentes. Il existe plusieurs possibilités pour animer un maillage : l'utilisation d'un squelette comme moyen de contrôle, en interpolant entre différentes postures clés définies manuellement ou encore en utilisant l'animation de marqueurs capturés.

**Déformation guidée par un squelette** La méthode la plus connue appelée *linear skin blending* (LBS) ou encore *skinning lisse* [105], consiste à lier une ou plusieurs articulations à chaque sommet du maillage, puis à déformer le maillage à partir de l'animation du squelette. Ce principe est simple et rapide, mais produit des déformations peu naturelles aux alentours des articulations, notamment lors des pliages et des torsions. En effet l'espace interpolé entre la rotation de deux articulations est compressé, ce qui engendre une perte de volume local. Ce phénomène classique est connu sous le nom de *collapsing-elbow effect* (voir figure 9.2). Ces inconvénients peuvent être considérablement améliorés en utilisant une interpolation des rotations par les quaternions duaux [93] (cf. figure 9.15). D'autres techniques plus complexes tiennent compte



FIGURE 9.2 – Illustration du *collapsing elbow* : perte de volume au niveau du coude.

des caractéristiques anatomiques (par exemple avec des modèles multicouches [26]) ou encore en utilisant des méthodes géométriques (par exemple par l'utilisation de *courbe de squelette* [188]). Budd [26] propose un modèle qui combine une animation par skinning classique couplée à une déformation basée sur le Laplacien autour des articulations. Cela a pour effet de mieux préserver le volume. Récemment McAdams [119] propose une méthode de déformation de maillage pour personnages articulés appliquée à des modèles élastiques. Le système permet par ailleurs de gérer les collisions en temps réel.

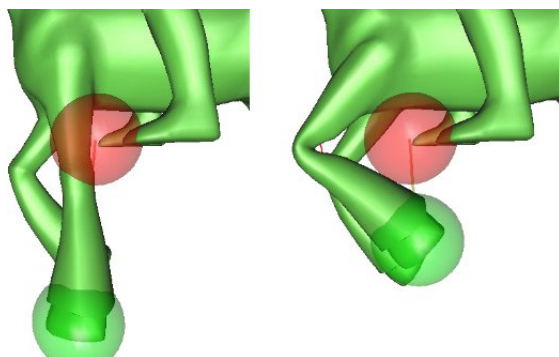


FIGURE 9.3 – Exemple de contrainte angulaire (Shi [149]).

Huang et col. [83] proposent en 2006 une approche proche de la notre, qui préserve lors de l'édition des contraintes de rigidité liées au squelette ainsi que le volume. Ces contraintes étant non linéaires, les auteurs proposent de résoudre le problème par une descente de gradient dans un sous espace simplifié (maillage de contrôle simplifié couvrant le maillage de référence). Les contraintes induites par le squelette et la préservation du volume sont projetées sur le maillage de contrôle par la méthode d'*interpolation de la valeur moyenne* (*mean value interpolation*). L'utilisation de ce sous espace améliore la rapidité de la méthode et rend le système plus stable. Zhao étend plus récemment la méthode en utilisant un maillage de contrôle te-

traédral en utilisant la méthode de projection *Modified Barycentric Interpolation* ([193]). Une autre étude menée par Shi et col. [149] (figure 9.3) combine dans un même *framework* les deux approches de déformation de maillage et d'animation par *skinning*. D'un côté ils définissent une contrainte de préservation des détails de la géométrie par une représentation Laplacienne du maillage [156] puis établissent le lien entre le maillage et le squelette à travers une phase de *rigging* classique. De l'autre, leur système préserve certaines propriétés du squelette durant la déformation telles que le bon respect des longueurs des segments, une contrainte de basculement (le centre de gravité du modèle doit être au dessus de sa surface de sustentation) et des contraintes de distance entre certaines articulations permettant de respecter approximativement certaines butées angulaires. Son système repose également sur une descente de gradient alternant (i) une estimation de la surface de la géométrie (ii) puis à partir de celle ci une résolution des contraintes liées au squelette à travers le lien de *rigging*. Plus récemment Jacobson et co-auteurs [85] utilisent des poids biharmoniques bornés pour déformer intuitivement des géométries associées à une structure de squelette. Ces trois dernières études permettent un contrôle multiple du maillage (par le squelette ou par des points de contrôles) et produisent des animations de maillage guidées par l'animation d'un squelette ou en interpolant entre des postures clés définies manuellement. Cependant elles n'ont pas été testées sur des séquences animées guidées par des données capturées.

Dans un autre contexte, Capell et ses co-auteurs [30] proposent un système basé sur de la simulation physique guidée par l'animation du squelette. Les positions des articulations sont intégrées à une structure volumique simplifiée qui sert de contrôle sur laquelle est appliquée une méthode des éléments finis.

Bien que l'ensemble des techniques décrites dans ce paragraphe intègre des informations liées au squelette, elles le font de façon explicite (c'est le squelette qui contrôle le maillage). Ainsi elles nécessitent une reconstruction de l'animation du squelette à partir de données capturées, ce que nous cherchons à éviter ici. De plus, certaines techniques nécessitent d'effectuer au préalable une étape de *rigging* peu évidente, le plus souvent opérée manuellement, ainsi qu'une mise en correspondance entre la géométrie et le squelette : la T-pose. Toutes ces contraintes rendent par ailleurs le transfert de l'animation d'un modèle vers un autre compliqué.

**Animation guidée par les marqueurs** D'autres études permettent d'animer des modèles géométriques par l'animation des marqueurs. Dans ce contexte, Zordan et co-auteurs [197] proposent un système dynamique entraîné par le mouvement des marqueurs capturés, ce qui garantit plusieurs propriétés physiques : la rigidité globale du système, la préservation de la longueur des segments et la prise en compte des limites angulaires des articulations. Cette approche est proche de la notre, puisque elle poursuit le même but, mais utilise un système physique alors que notre méthode s'applique à un système géométrique. Dans un tout autre registre Kravevoy & col. [100] proposent une méthode pour retrouver l'animation d'un maillage directement à partir des marqueurs sans information liée au squelette. Après une correspondance opérée manuellement entre les données capturées et le maillage, les auteurs utilisent une

technique basée sur une descente de gradient appelée *Mean-value Encoding* qui encode pour chaque sommet sa relation spatiale avec ses voisins dans un plan. Cette représentation améliore la stabilité et la vitesse de résolution de la méthode. Cette dernière, qui est applicable uniquement sur des géométries différentiables, est efficace pour des mouvements grossiers. Comme les auteurs l'indiquent, la méthode ne prend pas en considération l'information liée au squelette et ne permet donc pas de préserver les propriétés de rigidité et les limites angulaires des articulations.

Park et co-auteurs [136] (figure 9.4) ont également proposé une technique pour animer des maillages à partir de marqueurs issus de la capture de mouvement. Le lien entre le maillage et les marqueurs est réalisé en établissant une relation de type *skinning* obtenue par la segmentation de zones semi-rigides du maillage. Ensuite une structure surfacique intermédiaire fait le lien entre les marqueurs pour retrouver les trajectoires des marqueurs manquants. Les résultats de l'animation, qui intègrent les variations de géométrie induites par les déformations des muscles, sont très réalistes. Cependant, leur méthode nécessite un grand nombre de marqueurs (plusieurs centaines). Au contraire, la méthode proposée par Chai et co-auteurs [34] propose l'animation d'un maillage à partir d'un faible nombre de points de contrôle, grâce à un espace d'apprentissage déterminé à partir de plusieurs échantillons de maillages animés. Notre approche est très similaire à celle proposée par Stoll et co-auteurs [163]. En effet, les auteurs utilisent également un maillage de contrôle qu'ils découpent volumétriquement avec des tétraèdres. La méthode alterne une technique de déformation Laplacienne et un processus de mise à jour des différentielles Laplaciennes par une descente de gradient. Leur système est stable même pour de grandes déformations et produit des animations pour des maillages de haute qualité à partir de données capturées. Bien que prometteuse, cette méthode ne tient pas compte de certaines propriétés du squelette telles que la longueur des segments ou les limites angulaires des articulations, ce que nous considérons comme essentiel pour garantir des déformations naturelles.



FIGURE 9.4 – Illustration de la méthode de Park et col. [136]. À gauche le partitionnement du maillage en fonction du placement des marqueurs. Au milieu la structure intermédiaire faisant le lien entre les marqueurs. À droite la déformation du maillage obtenu.

Ref.	approche	animation	représentation volu- mique	cage	squelette	limites angulaires
Capell [30]	physique	oui (g.s)	oui	non	oui	non
Sorkine [156]	géom. (o.l)	non	non	non	non	non
Zordan [197]	physique	oui (g.m)	non	non	oui	oui
Botsch [22]	géom. (o.l)	non	non	non	non	non
Krayevoy [100]	géom. (o.l)	oui (g.m)	non	non	non	non
Yu [189]	géom. (o.l)	non	non	non	non	non
Zhou [194]	géom. (o.l)	oui (g.c)	oui	non	non	non
Chai [34]	apprentissage	oui (g.m)	non	non	non	non
Sumner [164]	apprentissage	oui (i.p.c)	non	non	non	non
Huang [83]	géom. (o.l)	oui (i.p.c)	oui	oui	oui	non
Lu [113]	géom.	oui (i.p.c)	non	non	non	non
Park [136]	géom. (skinning)	oui (g.m)	non	non	non	non
Shi [149]	géom. (o.l)	oui (i.p.c)	non	non	oui	approx.
Sorkine [160]	géom. (ARAP)	non	non	non	non	non
Song [155]	géom.	non	oui	non	non	non
Stoll [163]	géom. (o.l)	oui (g.m)	oui	oui	rigidité	non
Huang [82]	géom.	non	oui	oui	non	non
Martin [116]	physique	non	non	non	non	non
Zhao [192]	géom. (o.l)	non	oui	oui	non	non
Borosan [21]	géom. (ARAP)	non	non	oui	non	non
Zhang [190]	géom. (ARAP)	non	non	oui	approx.	non
Chao [35]	géom.	non	oui	non	non	non
Manson [115]	géom. (ARAP)	non	non	non	non	non
Jacobson [85]	géom. (o.l)	oui (i.p.c)	oui	non	oui	non
Zhao [193]	géom. (o.l)	non	oui	oui	non	non
Faraj [54]	géom. (ARAP)	non	non	oui	non	non
Dey [48]	géom. (o.l)	non	non	non	approx.	non

TABLE 9.1 – Approches connexes dédiées à l’animation guidée par les marqueurs ou à la déformation de maillage (présentées dans l’ordre chronologique). o.l : méthodes basées sur l’opérateur Laplacien. g.s : guidée squelette, g.m : guidée marqueurs, g.c : guidée par des courbes animées, i.p.c : interpolation entre postures clés et a.p : animation physique guidée par des forces.

### 9.2.3 Notre approche

La plupart des méthodes de déformation de maillage présentées dans la section précédente intègrent des contraintes de rigidité et de volume permettant de produire des déformations de qualité. Elles sont toutefois uniquement dédiées à la génération de postures statiques ou encore pour de l'animation à partir d'interpolation de postures définies manuellement [83, 85, 113]. Deux études proposent d'animer un maillage directement avec des marqueurs [100, 163] mais n'intègrent pas ou que partiellement les propriétés liées au squelette. Par ailleurs, aucune des méthodes proposées n'introduit les notions de limites angulaires liées aux articulations. Pourtant nous sommes convaincu que cette notion est essentielle pour générer des configurations du maillage cohérentes. Nous pensons que cette propriété est indispensable pour de l'animation et c'est en particulier le cas pour les mains qui possèdent de nombreuses articulations et beaucoup de contraintes associées aux limitations angulaires (dans la vidéo jointe, il est facile d'observer dans l'animation des mains que l'absence de butées angulaires produit des postures non naturelles). Seul Zordan [197] a proposé d'introduire les limites angulaires dans le contexte d'un système physique.

Notre objectif principal est d'animer une géométrie directement par l'animation des marqueurs, tout en préservant les détails de la géométrie. Notre approche s'inspire donc logiquement des techniques de géométrie différentielle récentes. De plus afin de préserver la rigidité à l'intérieur du maillage, nous utilisons une structure volumique. Cela permet de réduire les inter-pénétrations du maillage, ainsi que les artefacts autour des articulations et de mieux préserver le volume. Notre système étant dédié à de l'animation de personnages, il est impératif de préserver la morphologie du maillage. Nous étendons donc les modèles précédents en incorporant dans notre structure volumique un squelette décrit par un vecteur d'angles et de position. Cette structure nous permet (i) de préserver la rigidité de certaines zones (zones inter-articulaires) et de préserver la longueur des segments. (ii) de respecter des limites angulaires lors de la déformation.

## 9.3 Présentation du système

Notre méthode a pour objectif d'animer et de déformer des corps articulés. Notre principal objectif est de guider la déformation du maillage par les marqueurs mais notons également que notre système permet de l'animer par un contrôle du squelette ou de le déformer par une édition manuelle. L'étude qui suit s'applique indifféremment à la nature des données de contrôle.

Dans la figure 9.5, une illustration du processus est donnée pour un exemple d'animation de main guidée par des marqueurs capturés. Ces marqueurs sont positionnés à des emplacements spécifiques sur ou au-dessus de la surface de la géométrie avec aucune hypothèse sur leur positionnement vis-à-vis des articulations. A partir d'une géométrie de référence qui peut être fortement détaillée (et qui peut présenter une différence morphologique importante avec le modèle de main capturé), un maillage de contrôle volumique est créé. Ce maillage com-

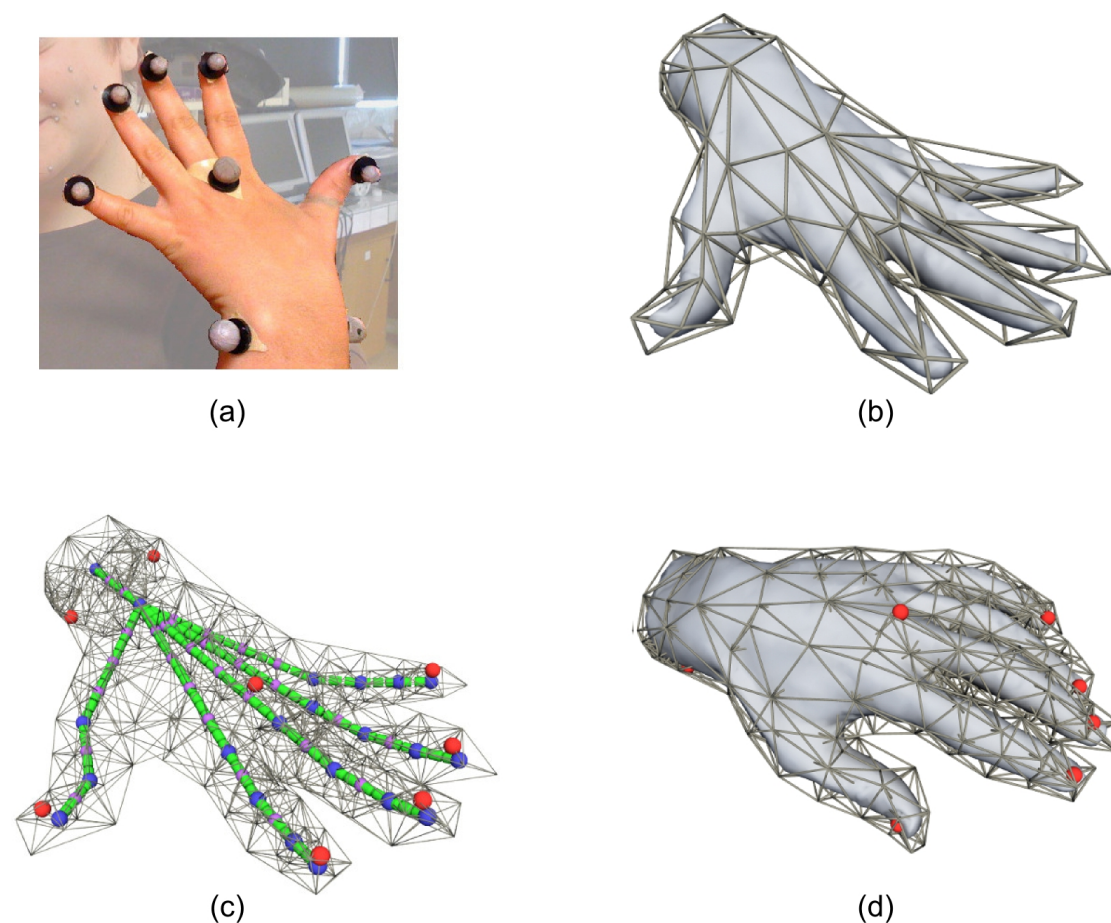


FIGURE 9.5 – Les différentes étapes de notre méthode. (a) : la main avec les marqueurs optiques. (b) : le maillage de référence et son maillage *cage*. (c) : le maillage de contrôle intégrant les données du squelette et des marqueurs positionné conformément aux marqueurs originaux de la main. (d) : une posture produite par le système.

porte une structure géométrique simplifiée enveloppant le maillage original, les informations liées au squelette et éventuellement des tétraèdres supplémentaires associant les marqueurs à la géométrie.

Notre système déforme ensuite le maillage de référence en deux phases : *i*) premièrement le maillage de contrôle est déformé en préservant la relation spatiale entre les positions des marqueurs animés et le maillage de contrôle ; cette déformation tend à préserver la forme originale du maillage de contrôle tout en prenant en considération les contraintes de rigidité induites par le squelette. *ii*) ensuite la déformation du maillage de contrôle est directement appliquée à la géométrie de référence (de haute qualité) via la technique des *Green Coordinates* [108].



## 9.4 Déformation du maillage volumique incluant les propriétés du squelette

Dans cette section, nous détaillons les différentes opérations de notre processus d'animation. Nous commencerons par expliquer comment le maillage de contrôle est construit. Nous présenterons ensuite l'application de notre méthode présentée au chapitre 6 à la déformation du maillage volumique, avec en particulier la préservation des propriétés du squelette. Enfin, nous montrerons comment déformer le maillage de référence avec la technique des *Green Coordinates*, avant de donner une vue d'ensemble sur l'algorithme final.

### 9.4.1 Couplage squelette/maillage via la construction du maillage volumique

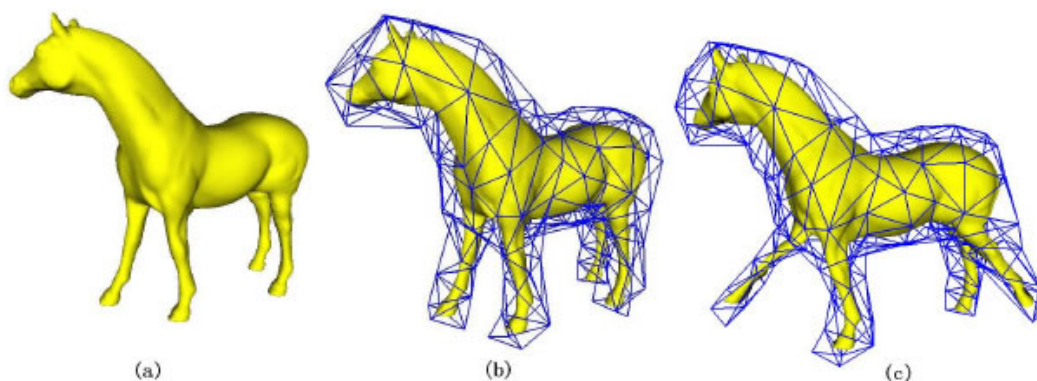


FIGURE 9.6 – Exemple de déformation par cage. (a) le modèle *cheval* de référence ; (b) en fil de fer bleu le maillage cage ; (c) Le résultat d'une déformation. Illustration tirée de [184]

**Création du squelette** En ce qui concerne la construction du squelette, plusieurs méthodes ont été proposées pour le créer automatiquement à partir du maillage de référence, *e.g* [9]. Il est également possible de le construire manuellement avec un logiciel commercial comme Maya, Motionbuilder ou 3DSMax. Cette étape prend quelques minutes et ne nécessite pas de compétences spécifiques. Sa construction établie, il n'existe pas de relation explicite entre le maillage et le squelette en dehors de leur position relative dans l'espace 3D (le squelette doit être placé à l'intérieur du maillage).

**Création du maillage cage** Le maillage volumique est ensuite créé en définissant une structure intermédiaire, généralement connue sous le nom de *maillage cage* [121]. Les *maillages cages* sont des structures simplifiées recouvrant le maillage de référence. Le principe d'une déformation par *cage* est d'utiliser ce maillage simplifié comme maillage de contrôle (celui

édité par l'utilisateur) et d'appliquer sa déformation au maillage de référence. La figure 9.6 illustre bien le principe. Il existe d'ailleurs plusieurs méthodes permettant d'établir le lien entre les deux maillages [88, 89, 108]. Notez que pour des structures complexes telles que les mains, ce maillage cage doit préserver au mieux la morphologie du modèle de référence comme par exemple les doigts. Ainsi il convient de prendre en compte plus de détails autour des articulations afin de mieux préserver les caractéristiques géométriques lors des grandes déformations. Il existe un certain nombre de méthodes pour le créer automatiquement comme celle proposée par [184]. Pour notre application, nous avons simplement appliqué une simplification du maillage par effondrement des arêtes (appelée *edge collapse*) suivie d'un agrandissement général du maillage permettant un recouvrement du maillage de référence. Le maillage cage résultant est caractérisé par l'absence de trous ou d'intersections.

**Le maillage volumique** Afin de créer le maillage volumique (aussi appelé le maillage de contrôle dans la suite de ce document), nous utilisons une tétraédrisation de Delaunay appliquée au maillage cage surfacique. Cette tétraédrisation comprend également :

- l'ensemble des positions des marqueurs. Ces marqueurs sont positionnés manuellement aux bons emplacements vis-à-vis de la géométrie de référence (souvent à quelques millimètres à l'extérieur) et liés au maillage de contrôle par la tétraédrisation,
- les sommets et les arêtes correspondant à une discrétisation du squelette (points bleus et arêtes vertes dans la figure 9.5). Cette discrétisation est effectuée par un échantillonnage spatial régulier du squelette. La valeur de l'échantillonnage des segments du squelette permet de contrôler le nombre de sommets associés à l'ossature dans le processus de tétraédrisation. Ainsi nous le considérons comme un paramètre définissant la complexité de calcul global.

La tétraédrisation de Delaunay est un processus complexe, ainsi plusieurs travaux ont été proposés à ce sujet [25, 45, 148]. Nous avons choisi d'utiliser la méthode développée par Miller et ses collaborateurs [124], appelée *piecewise linear complex (PLC)*, et mise en œuvre par Si et co-auteurs [151]. Cette méthode présente les avantages d'être robuste et flexible. Finalement, le maillage obtenu est fermé et composé de triangles, ce qui lui confère des propriétés permettant d'appliquer la méthode définie ci-après.

#### 9.4.2 Déformation de maillage As-Rigid-As-Possible

Nous présentons dans cette section notre processus de déformation du maillage volumique. Nous commençons d'abord par rappeler les principes de base de l'édition de surface par l'opérateur Laplacien appliqué aux maillages, avant de présenter une adaptation de l'algorithme *As-Rigid-As-Possible* à une structure volumique. Dans cette section nous utilisons les concepts d'édition de graphe présentés en 6 et étendons ceux dédiés à l'édition de maillage par la technique *As-Rigid-As-Possible* présenté dans l'annexe B.

**Édition Laplacienne de surface (LSE)** Le concept d'édition de maillage surfacique est similaire à celui de l'édition de graphe (présenté dans le paragraphe 5.2.2) appliqué à des maillages. Ainsi nous considérons une structure triangulée fermée (ce qui est le cas du maillage de contrôle), caractérisée par  $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ , où  $\mathbf{E}$  décrit la connectivité, et  $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  est le vecteur de coordonnées Euclidiennes des sommets du maillage. Rappelons qu'à partir de  $\mathbf{V}$  et de l'ensemble  $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_m\}$  de points de contrôle, nous pouvons calculer les nouvelles positions  $\mathbf{V}' = \{\mathbf{v}'_1, \dots, \mathbf{v}'_n\}$  qui minimisent la fonction quadratique donnée par

$$E(\mathbf{V}') = E_l(\mathbf{V}') + E_u(\mathbf{V}'). \quad (9.1)$$

Le premier terme  $E_l$  pénalise les différences entre les coordonnées différentielles avant et après reconstruction et le second terme  $E_u$  pénalise les modifications des positions des points de contrôle.

L'équation 9.1 peut être ensuite minimisée comme un système linéaire de dimension  $(n + m) \times n$  sur-déterminé

$$\begin{pmatrix} \mathbf{L} \\ \mathbf{I}_u \end{pmatrix} \mathbf{V}' = \begin{pmatrix} \Delta \\ \mathbf{U} \end{pmatrix} \quad (9.2)$$

où  $\mathbf{I}_u$  est la matrice d'indices associée aux contraintes de position comprises dans la matrice  $\mathbf{U}$ . La forme géométrique reconstruite paraît généralement naturelle tant que les déformations impliquant des rotations sont de faibles amplitudes. Comme nous l'avons vu précédemment (section 5.3.1.1), pour de plus grandes transformations, Sorkine [159] propose en 2007 une méthode de résolution non linéaire appelée *ARAP* (as-rigid-as-possible) pour préserver localement la rigidité de chaque cellule du maillage. Dans les paragraphes suivants nous rappelons les bases de *ARAP* surfacique puis l'étendons à des maillages volumiques.

**ARAP surfacique** Rappelons le principe général de la méthode surfacique. A partir d'une première estimation de  $\mathbf{V}'$  donnée par l'équation 9.1, le but d'*ARAP* est de trouver itérativement les nouvelles valeurs des coordonnées différentielles préservant la rigidité locale. Ainsi, si la cellule  $C_i$  du maillage correspondant au sommet  $i$  est déformée en une cellule  $C'_i$ , la transformation rigide approximative entre  $C_i$  et  $C'_i$  peut être décrite par la matrice de rotation qui minimise

$$E(C_i, C'_i) = \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)\|^2, \quad (9.3)$$

où  $\mathcal{N}(i)$  est l'ensemble de sommets connectés au sommet d'indice  $i$  et  $w_{ij}$  est le poids cotangent associé aux sommets  $i$  et  $j$ . Le problème peut ensuite être réécrit comme un problème itératif,

chaque itération étant exprimée par  $E(\mathbf{V}') = E_a(\mathbf{V}') + E_u(\mathbf{V}')$ , avec  $E_a$  défini par :

$$E_a(\mathbf{V}') = \sum_{i=1}^n \|\delta_i^A - \mathcal{L}(\mathbf{v}_i')\|^2 \quad (9.4)$$

où  $\delta_i^A$  est le terme évalué à chaque itération. Dans la suite de ce document, nous utilisons la notation  $\Delta^A = \delta_i^A$ . La méthode ARAP est davantage détaillée dans l'annexe B.

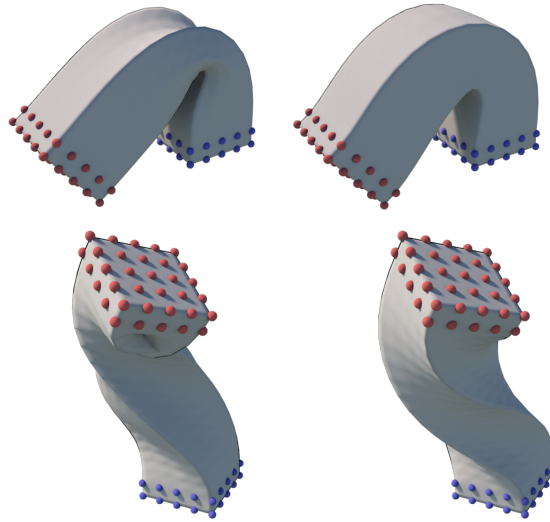


FIGURE 9.7 – Exemple de déformation arbitraire avec les méthodes *ARAP* surfacique (à gauche) et volumique (à droite) à l'issue de 100 itérations. La méthode volumique limite les intersections inter-maillage et préserve mieux le volume. (Erreurs relatives des volumes : *ARAP* surfacique haut : 0.072, bas : 0.2205, *ARAP* volumique haut : 0.018, bas : 0.1345).

**ARAP volumique** L'algorithme précédent est efficace pour des maillages surfaciques mais peut être amélioré en l'étendant à des structures volumiques. Nous proposons dans cette section une version volumique de la technique *ARAP*. La cellule  $C_i$  associée à un sommet  $i$  peut être considérée comme un élément de variété de dimension 3, et en tant que telle son énergie est liée non seulement à son volume, mais aussi à sa forme globale. Tout comme pour des maillages surfaciques, le processus de tétraédrisation garantit que le maillage est différentiable dans son intégralité. L'opérateur de Laplace Beltrami est donc applicable sur l'ensemble du maillage. Une possibilité pour calculer les poids de l'opérateur de Laplace-Beltrami est de continuer à utiliser les poids cotangents pour équilibrer la contribution de l'ensemble des sommets de la cellule (en choisissant arbitrairement les triangles plans partageant l'arête par exemple). Toutefois, comme le suggère Zhou [194] dans un autre contexte, nous optons pour l'utilisation

d'un noyau Gaussien exprimé par :

$$w_{ij} = \exp\left(-\frac{|\mathbf{v}_i - \mathbf{v}_j|^2}{\gamma}\right). \quad (9.5)$$

En effet, cette paramétrisation donne une bonne approximation de l'opérateur de Laplace-Beltrami [40] quand  $\gamma \rightarrow 0$ . Par ailleurs, le paramètre  $\gamma$  peut être automatiquement défini comme la distance médiane des voisins les plus proches.

Le poids  $w_{ij}$  étant un scalaire, son utilisation dans *ARAP* est similaire à la méthode originale. En effet nous pouvons reprendre la démarche proposée par Sorkine en appliquant les poids à noyau Gaussien à la place des poids cotangents. A partir de l'équation 9.3, l'énergie globale du maillage à minimiser est définie par la fonction quadratique :

$$E(\mathcal{M}') = \sum_{i=1}^n E(C_i, C'_i) = \sum_{i=1}^n \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)\|^2 \quad (9.6)$$

où  $\mathcal{M}'$  est le maillage déformé et dans notre cas  $w_{ij}$  est le poids associé à l'arête  $e(ij)$  suivant le noyau Gaussien. Pour trouver les positions des sommets transformés  $\{\mathbf{v}'_i\}$  de  $\mathcal{V}'$  qui minimisent  $E(\mathcal{M}')$ , rappelons que la méthode ARAP propose d'utiliser une stratégie de minimisation alternative. Ainsi à partir d'un ensemble de transformations rigides  $\{\mathbf{R}_i\}$ , nous trouvons les positions qui minimisent  $E(\mathcal{M}')$ . Puis à partir de l'ensemble des positions  $\{\mathbf{v}'_i\}$ , nous calculons les transformations rigides  $\{\mathbf{R}_i\}$  qui minimisent  $E(\mathcal{M}')$  et ainsi de suite. Nous avons détaillé dans l'annexe B le calcul de l'ensemble des rotations  $\{\mathbf{R}_i\}$ . Pour calculer les positions  $\{\mathbf{v}'_i\}$  optimales à partir des rotations, nous déterminons l'équation pour laquelle le gradient de  $E(\mathcal{M}')$  en fonction des  $\{\mathbf{v}'_i\}$  s'annule.

$$\begin{aligned} \frac{\partial E(\mathcal{M}')}{\partial \mathbf{v}'_i} &= \frac{\partial}{\partial \mathbf{v}'_i} \left( \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)\|^2 + \sum_{j \in \mathcal{N}(i)} w_{ji} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_j(\mathbf{v}_i - \mathbf{v}_j)\|^2 \right) \\ &= \sum_{j \in \mathcal{N}(i)} 2w_{ij} ((\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)) + \sum_{j \in \mathcal{N}(i)} 2w_{ji} ((\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_j(\mathbf{v}_i - \mathbf{v}_j)). \end{aligned}$$

En utilisant  $w_{ij} = w_{ji}$ , nous obtenons

$$\frac{\partial E(\mathcal{M}')}{\partial \mathbf{v}'_i} = \sum_{j \in \mathcal{N}(i)} 4w_{ij} ((\mathbf{v}'_i - \mathbf{v}'_j) - \frac{1}{2}(\mathbf{R}_i + \mathbf{R}_j)(\mathbf{v}_i - \mathbf{v}_j)). \quad (9.7)$$

Puis en posant que la dérivée partielle est égale à zéro pour chaque  $\mathbf{v}'_i$ , nous obtenons le système d'équations linéaire éparse :

$$\sum_{j \in \mathcal{N}(i)} w_{ij} (\mathbf{v}'_i - \mathbf{v}'_j) = \sum_{j \in \mathcal{N}(i)} \frac{w_{ij}}{2} (\mathbf{R}_i + \mathbf{R}_j)(\mathbf{v}_i - \mathbf{v}_j). \quad (9.8)$$

Finalement en remarquant que le premier terme est équivalent à l'opérateur de Laplace Beltrami discret associé au sommet  $i$ , nous pouvons généraliser la démarche à l'ensemble du maillage par une notation plus simplifiée :

$$E_a(\mathcal{M}') = \sum_{i=1}^n \|\delta_i^A - \mathcal{L}(v_i')\|^2 \quad (9.9)$$

où  $\delta_i^A$  est le terme évalué à chaque itération.

Les résultats obtenus sont illustrés par les figures 9.7 et 9.8. L'apport d'une structure volumique rend la méthode plus robuste : elle prévient davantage les intersections inter-maillage et préserve mieux le volume. Bien que cette méthode donne des résultats efficaces, elle ne permet pas une bonne conservation des propriétés du squelette : la rigidité autour des segments, longueur des segments et limites angulaires des articulations. Dans la section suivante, nous montrons comment considérer ces contraintes implicites dans le processus de déformation.

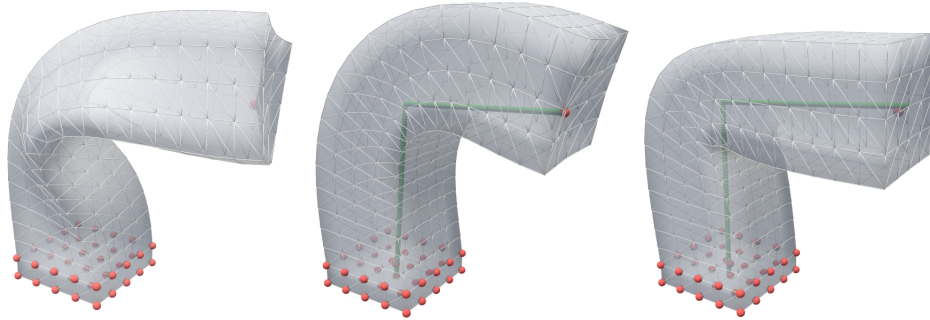


FIGURE 9.8 – Exemple de déformation avec ARAP surfacique (à gauche) et volumique (au milieu et à droite) après 100 itérations. La représentation volumique plus robuste permet de préserver la forme originale du maillage. Le modèle de droite est le résultat de notre méthode incorporant les contraintes liées au squelette.

### 9.4.3 Préservation des propriétés du squelette

La structure du squelette est comprise dans la tétraédrisation : les positions des articulations sont représentées par un ensemble de sommets et les segments entre les articulations sont décomposés en un ensemble d'arêtes (figure 9.5). L'ensemble de ces informations est inclus dans le maillage de contrôle. Pour garantir des animations réalistes avec une configuration des articulations cohérente, deux propriétés inhérentes au squelette sont à respecter. Tout d'abord, nous introduisons une contrainte permettant la préservation des longueurs des arêtes du maillage ainsi que la colinéarité entre les arêtes d'un même segment. Comme illustré dans la figure 9.8 (modèle de droite), cette contrainte a pour objectif de donner une déformation rigide suivant la structure de squelette. Deuxièmement, nous proposons d'ajouter des limites angulaires aux articulations. Cependant ces nouvelles contraintes ne sont pas linéaires, et les limites angulaires

ne sont pas directement applicables avec notre formulation du problème (pas de représentation angulaire).

Pour résoudre ces deux problèmes, nous nous proposons donc d'étendre notre méthode d'édition introduite dans la section 6. Rappelons que notre méthode propose de préserver les relations spatiales entre les sommets d'une structure par un couplage entre l'information différentielle du sommet avec ses voisins et éventuellement sa distance avec d'autres sommets. Notre méthode se décompose en deux fonctionnelles : l'une linéaire minimisant l'information différentielle des sommets, l'autre non linéaire préservant les contraintes de distance entre les sommets. Appliquée à notre problème, la première minimisation est associée au critère défini dans l'équation 9.9. Nous proposons de détailler dans le paragraphe suivant la manière d'intégrer les propriétés du squelette sous la forme de contraintes différentielles.

**Intégration des contraintes du squelette** Décrivons dans un premier temps les objectifs de notre technique de minimisation alternée appliquée à notre problème. Comme décrit précédemment, cette technique fonctionne de manière alternée : dans un premier temps (i) nous calculons une déformation approximative par la méthode *As-Rigide-As-Possible* volumique, ce qui nous donne un bon pré-conditionnement du maillage volumique. (ii) nous calculons ensuite une configuration acceptable du squelette respectant les contraintes liées aux segments et aux limites angulaires. Enfin, (iii) nous intégrons cette nouvelle configuration du squelette au moyen d'une contrainte linéaire dans le problème *As-Rigide-As-Possible* volumique. Ces deux dernières étapes sont répétées jusqu'à convergence. Nous l'avions mentionné dans la section 6.2.2, cette contrainte s'exprime par un système linéaire au sens des moindres carrés :

$$DV' = \Gamma \quad (9.10)$$

où  $D$  est la matrice de dimension  $\text{card}(E_S) \times n$  dont le rôle est de calculer les coordonnées différentielles de chaque segment  $\in E_S$ . Ici  $E_S$  est l'ensemble des arêtes constituant les segments du squelette. Notre objectif est donc de calculer le vecteur de différentielles  $\Gamma$  estimant la plus proche configuration d'un squelette acceptable pré-conditionné par l'équation 9.9.

**Calcul de  $\Gamma$**  Les coordonnées différentielles stockées dans  $\Gamma$  sont déduites à partir d'une configuration du squelette correcte respectant les contraintes de rigidité ainsi que les limites angulaires. Ces dernières sont fixées au préalable par l'utilisateur. Ce résultat est obtenu grâce à une procédure répartie sur deux phases :

(i) Premièrement, la posture approximative du squelette courant  $S_c$  est extraite à partir des sommets du maillage de contrôle correspondant aux articulations du squelette. Le squelette est parcouru depuis l'articulation *racine* vers les articulations filles et pour chaque segment représenté par le vecteur  $\vec{\alpha}_{seg}$ , nous ré-estimons son orientation en fonction de l'ensemble des

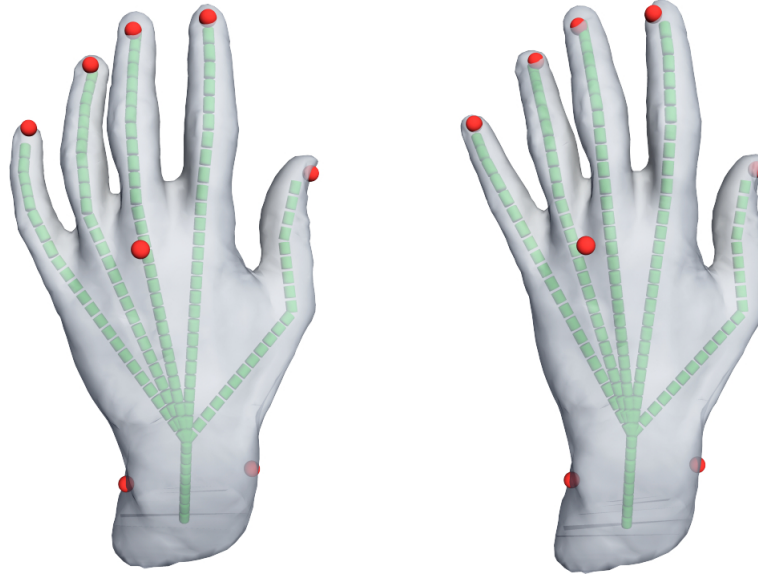


FIGURE 9.9 – Exemple de posture de mains montrant l’influence des limites angulaires des articulations pour un même maillage de contrôle et les mêmes marqueurs cibles : à gauche sans limites articulaires et à droite avec limites articulaires.

arêtes le constituant associé au bon respect de sa longueur initiale :

$$\vec{\alpha}_{seg} = l_{ab} \frac{1}{k} \sum_{e_{ab} \in seg} \vec{e}_{ab} \quad (9.11)$$

où  $k$  est le nombre d’arêtes constituant le segment,  $\vec{e}_{ab}$  est le vecteur représentant l’arête  $e_{ab}$  et  $l_{ab}$  sa longueur initiale. Ce squelette intermédiaire a ses segments de bonnes longueurs, mais ne respecte pas nécessairement les limites angulaires.

(ii) Nous projetons ensuite  $S_c$  dans l’espace des postures admissibles par une simple correction des valeurs d’angles articulaires en choisissant les valeurs acceptables les plus proches (ce travail est présenté dans le paragraphe suivant). Cette dernière étape produit un nouveau squelette  $S_a$ .

$\Gamma$  est alors calculé directement à partir de  $S_a$  par le calcul direct des vecteurs  $\gamma_{ij}$ . La figure 9.9 illustre la déformation de la main avant et après l’ajout de contraintes. La posture de la main semble plus naturelle dans ce second cas, puisque l’ensemble des doigts respectent les limites angulaires à préserver. Ainsi les rotations anormales le long des doigts n’apparaissent plus.

#### 9.4.4 Projection du squelette dans l’espace des postures admissibles

Une articulation est définie par sa position et son orientation. Dans la plupart des cas, elle possède 3 degrés de liberté (DoFs). Plusieurs modèles ont été proposés pour définir les axes des



rotations. Le plus connu est appelé le modèle de *rotule* ou *ball-and-socket*. Plus précisément l'orientation associée à un segment est décomposée en deux rotations : un terme appelé *rotationnel* (2 DoFs) qui déplace l'extrémité du segment vers sa position finale et un autre appelé *orientationnel* (1 DoF) qui représente la rotation de *twist* (rotation autour de l'axe du segment). Baerlocher [13] étudie plus en détail le modèle et propose une paramétrisation appelée *swing-and-twist* en opposition à la paramétrisation des angles d'Euler. Le système *FABRIK* [7] utilise également ce modèle en projetant l'espace d'atteignabilité du composant *rotationnel* dans un plan. Wilhelms et Van Gelder [181] proposent d'utiliser un espace d'atteignabilité défini par un cône prédéfini par l'utilisateur. La valeur de l'articulation est obtenue en projetant le segment de l'articulation sur le cône, assurant ainsi qu'elle est bien dans les bonnes limites. Blow et col. [19] présentent une approche similaire en utilisant une fenêtre d'atteignabilité.

**Représentation des limites** Dans notre cas, nous utilisons également le modèle *rotule* qui donne une représentation intuitive à paramétrer. A partir de la posture de référence du squelette, nous calculons dans un premier temps les bases associées à chaque articulation. Par exemple à partir des positions  $\mathbf{p}_0$ ,  $\mathbf{p}_1$  et  $\mathbf{p}_2$  de 3 articulations, nous définissons la base orthonormée  $\mathbf{B}'_1$  d'axes  $\mathbf{x}'_1$ ,  $\mathbf{y}'_1$  et  $\mathbf{z}'_1$  de l'articulation  $a_1$  par :

$$\begin{cases} \vec{x}'_1 = \frac{\mathbf{p}_2 - \mathbf{p}_1}{\|\mathbf{p}_2 - \mathbf{p}_1\|} \\ \vec{z}'_1 = \mathbf{x}'_1 \times \frac{\mathbf{p}_1 - \mathbf{p}_0}{\|\mathbf{p}_1 - \mathbf{p}_0\|} \\ \vec{y}'_1 = \mathbf{x}'_1 \times \mathbf{z}'_1 \end{cases}$$

$\mathbf{x}_1$  est le composant *orientationnel* alors que  $\mathbf{z}_1$  et  $\mathbf{y}_1$  sont les composants *rotationnels* de la rotation. Une illustration en 2D est donnée par la figure 9.11. Nous définissons ensuite les limites angulaires associées à chaque articulation manuellement pour chaque axe, soit un intervalle de définition pour chaque angle d'Euler de l'articulation. D'une manière générale, nous notons pour la suite de cette section la base dans la représentation *rotule* de l'articulation  $i$  par  $\mathbf{B}'_i$  dont les axes sont  $\mathbf{x}'_i$ ,  $\mathbf{y}'_i$  et  $\mathbf{z}'_i$ .

La figure 9.10 illustre bien l'intérêt de constituer de nouvelles bases pour définir les limites angulaires. Les axes  $\mathbf{y}$  et  $\mathbf{z}$  sont intuitifs à utiliser. Nous n'avons volontairement pas modifié le repère associé au noeud racine puisqu'il définit la rotation globale de la géométrie qui n'a pas lieu d'être limitée. A noter également que les repères associés à une articulation qui a plusieurs filles ont été dupliqués.

**Calcul des rotations induites par la déformation du maillage** L'objet de ce paragraphe est de montrer comment trouver les rotations produisant l'animation des articulations dans l'espace du squelette (bases  $\mathbf{B}'_i$ ) puis de formuler une correction de la posture courante en incluant les limites angulaires.

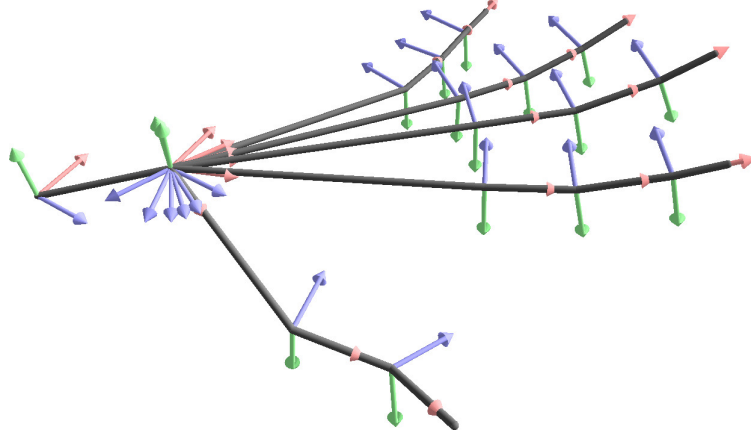


FIGURE 9.10 – Exemple des nouvelles bases associées à chaque articulation du squelette de la main.

Un squelette est un *arbre* tel que chaque articulation est exprimée par rapport à une articulation qui lui est directement parente (en dehors de la première articulation) (conférer figure 9.10). Rappelons que le calcul de la transformation (position et rotation) d'une articulation  $i$  est définie comme la combinaison de la transformation de l'articulation parente en absolu avec sa transformation relative :  $B'_i = B'_{i-1} * B_i^{Rel}$ . D'une manière générale  $B'_i = \prod_{k=0}^{k=i} B_k^{Rel}$  où  $k = 0$  correspond au noeud racine. En considérant un squelette de  $n$  articulations et un vecteur de rotations relatives  $\theta_i^{Rel} = \{\theta_0^{Rel}, \dots, \theta_n^{Rel}\}$  correspondant à la variation de rotation dans chaque base  $B'_i$ , la transformation absolue  $T_i^\theta$  associé à l'articulation  $i$  s'écrit :

$$T_i^\theta = \prod_{k=0}^{k=i} B_k^{Rel} * \theta_i^{Rel}. \quad (9.12)$$

L'algorithme actuel nous donne une configuration du squelette avec les positions des articulations dans l'espace absolu mais estime également les rotations  $R_i$  associées à chaque sommet du maillage données par l'algorithme de déformation ARAP. Nous avons donc proposé deux solutions pour retrouver les rotations dans les bases  $B'_i$ .

(i) **Détermination des  $\theta_i^{Rel}$  à partir des composants  $R_i$ .** Les rotations  $R_i$  définies dans l'équation 9.3 sont absolues. Nous pouvons donc déterminer leurs valeurs relatives en considérant leurs articulations parentes. La rotation relative  $R_i^{Rel}$  se calcule donc par  $R_i^{Rel} = R_{i-1}^{-1} * R_i$ .  $\theta_i$  est ensuite déterminé par un changement de repère tel que  $\theta_i^{Rel} = B_i^{-1'} * R_i^{Rel} * B_i'$ .  $\theta_i^{Rel}$  s'écrit donc :

$$\theta_i^{Rel} = (B_i')^{-1} * (R_{i-1}^{-1} * R_i) * B_i'. \quad (9.13)$$

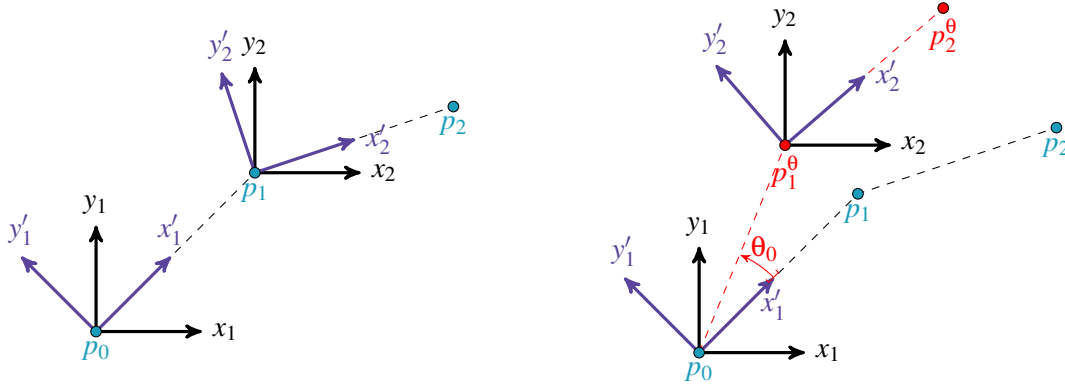


FIGURE 9.11 – Illustration de la définition des repères en 2D d'un squelette à 3 articulations. La figure de droite montre l'application d'un angle relatif à l'articulation 0.

Après correction de l'angle (expliquée dans le paragraphe suivant), cette solution préserve bien les limites angulaires toutefois les rotations  $\mathbf{R}_i$  sont trop approximatives et fournissent un mauvais pré-conditionnement à la correction du squelette. En effet ces rotations étant évaluées localement pour chaque sommet, elles ne fournissent qu'une valeur partielle de la rotation réelle. Il en résulte un suivi du mouvement grossier (plus de préservation des contacts par exemple) avec une mauvaise préservation des contraintes de position induites par les marqueurs.

(ii) **Détermination des  $\theta_i^{Rel}$  à partir des positions en absolu.** Notre seconde approche suit la méthode décrite et argumentée par Blow [19] qui consiste à calculer l'angle  $\theta_i^{Rel}$  à partir d'une configuration absolue du squelette par un quaternion. Une représentation du problème est illustrée par la figure 9.11. En considérant les deux vecteurs  $\mathbf{p}_{01} = \mathbf{p}_1 - \mathbf{p}_0$  et  $\mathbf{p}'_{01} = \mathbf{B}_i^{-1'} * (\mathbf{p}'_1 - \mathbf{p}'_0)$ , où  $\mathbf{p}'_{01}$  est le vecteur  $\mathbf{p}'_1 - \mathbf{p}'_0$  dans le repère de l'articulation  $i$ . Nous pouvons calculer le quaternion  $\mathbf{q}_0 = (\alpha_0, \alpha_0)$  tel que

$$\begin{cases} \alpha_0 = \frac{\mathbf{p}_{01}}{\|\mathbf{p}_{01}\|} \times \frac{\mathbf{p}'_{01}}{\|\mathbf{p}'_{01}\|} \\ \alpha_0 = \arccos\left(\frac{\mathbf{p}_{01} \cdot \mathbf{p}'_{01}}{\|\mathbf{p}_{01}\| \times \|\mathbf{p}'_{01}\|}\right) \end{cases}$$

**Correction de l'angle** Que ce soit la première ou la seconde approche, la correction de l'angle passe par une représentation de  $\theta_i^{Rel}$  par les angles d'Euler. Nous corrigeons finalement les angles en fonction des espaces de définition donnés précédemment avant de ré-introduire la rotation dans le calcul de la configuration du squelette en absolu :

$$\mathbf{T}_i^\theta = \prod_{k=0}^{k=i} \mathbf{B}_k^{Rel} * \theta_i^{Rel\_corrige}. \quad (9.14)$$

Un exemple du résultat est donné par la figure 9.9. Finalement notre méthode de déformation permet de retrouver les orientations des articulations au cours du temps dans un espace unique (dans la même base pour chaque articulation), ce qui nous permet donc de retrouver l'animation du squelette.

#### 9.4.5 Relâchement des contraintes d'ARAP autour des articulations

L'apport des contraintes liées au squelette sur-contraint fortement le problème de minimisation initialement défini. Contrairement à la méthode classique, notre nouveau problème de minimisation implique que l'énergie définie par l'équation (9.9) n'est plus répartie uniformément sur l'ensemble du graphe. Autour des zones proches des articulations, les transformations rigides associées aux cellules ont proportionnellement à l'importance de la pliure, des orientations divergentes. Cela amène lors de fortes déformations à des collisions inter-maillage.

Telle que définie, la méthode *ARAP* permet de trouver pour chaque sommet  $v_i$  du graphe une matrice de rotation  $R_i$ . Rappelons que cette matrice de rotation est considérée à l'issue de l'itération comme la meilleure approximation de la transformation rigide entre les deux cellules  $C_i$  et  $C'_i$  (respectivement les cellules avant et après déformation) associées aux sommets  $v_i$  et  $v'_i$ . En basant sur la déformation d'une cellule, le calcul de cette matrice prend logiquement en considération l'ensemble des voisins de  $v_i$ . Elle permet en particulier de préserver au mieux les relations spatiales entre  $v_i$  et ses voisins. Par ailleurs il est important de noter que lors d'une déformation rigide, c'est précisément aux endroits où se forment les pliures que ces relations spatiales sont le moins respectées. C'est pourquoi nous proposons lors du calcul de la transformation rigide  $R_i$  d'un sommet  $v_i$  d'établir une priorité entre les relations spatiales à préserver. Nous cherchons donc à relâcher les contraintes de rigidité locales à l'approche de ces zones en introduisant une priorité entre les cellules du graphe qui sont plus ou moins rigides (moins de rigidité proche des articulations). Nous avons proposé deux solutions pour résoudre ce problème.

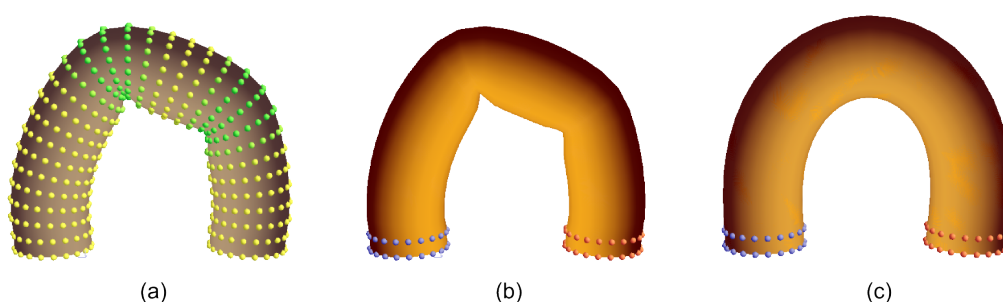


FIGURE 9.12 – En (a) et (b) la déformation obtenue avec notre méthode. En (a) la couleur des points indique le poids  $w^r$  de chaque sommet (jaune = 1, vert = 0). La même déformation sans contrainte de poids (c). (en bleu : les points fixes, rouge : les points de contrôle).

(i) **Poids définis manuellement** Nous proposons dans un premier temps une méthode ne

prenant pas en considération les contraintes du squelette. Elle est similaire au calcul des poids de skinning en donnant plus ou moins d'importance aux sommets en fonction de leurs distances avec le centre de l'articulation. Nous associons donc à chaque sommet du maillage un poids  $w_i^r = 1 - e^{-d_{ic}^2} \in [0, 1]$  où  $d_{ic}$  est la distance euclidienne entre le sommet  $i$  et le centre articulaire le plus proche du squelette :  $c$ . La figure 9.12 (a) illustre la diffusion des poids associés à chaque sommet pour un cylindre comportant quatre articulations (deux à l'intérieur et deux sur les bords). On définit donc une nouvelle formulation de la matrice de covariance lors du calcul de  $\mathbf{R}_i$  (défini dans l'annexe B) en fonction de  $w_i^r$  :

$$S_i = \sum_{j \in \mathcal{N}(i)} w_j^r w_{ij} e_{ij} e_{ij}'^T. \quad (9.15)$$

La figure 9.12 illustre bien le fait que les pliures apparaissent bien au niveau des articulations lors de la déformation. Néanmoins la méthode dépend fortement des paramètres utilisés pour la mesure de distance. Elle n'assure pas par ailleurs la notion de distance fixe entre les articulations qui est fondamentale puisque les segments du squelette sont normalement invariants à la déformation. Enfin malgré les pliures, la méthode n'assure pas pour autant une rigidité optimale entre les articulations.

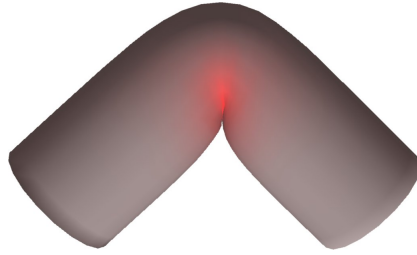


FIGURE 9.13 – Illustration de la répartition de l'erreur liée à la rigidité exprimée par l'Equation 9.3. En rouge proche de la pliure, la rigidité des cellules est moins respecté alors qu'en gris l'énergie est nulle.

**(ii) Poids définis automatiquement** L'erreur liée à la rigidité de chaque sommet est variable en fonction de la déformation opérée sur le graphe. Cette énergie est illustrée par la figure 9.13. Notons que cette répartition de l'énergie est directement liée au caractère sur-contraint du problème induit par les contraintes du squelette. Nous introduisons donc l'ajout d'un poids dynamique directement corrélé à l'énergie propre de chaque cellule calculée dans l'équation (9.3). La rigidité étant directement liée à la matrice  $\mathbf{R}_i$ , nous redéfinissons l'équation (B.6) (section 5.3.1.1) par :

$$S_i = \sum_{j \in \mathcal{N}(i)} \alpha_i w_{ij} e_{ij} e_{ij}'^T, \quad (9.16)$$

où

$$\alpha_i = \left( \frac{E(C_i, C'_i)}{\max(\mathcal{N}(E_{C_j}))} \right)^k, \quad (9.17)$$

où  $\mathcal{N}(E_{C_j})$  est l'ensemble des énergies des cellules du graphe et  $k$  est un paramètre donné manuellement.

La figure 9.14 compare les différents résultats obtenus avec (a) la méthode volumique non contrainte, (b) l'ajout de poids pour chaque sommet, (c) l'introduction de contraintes liées au squelette et (d) la combinaison des contraintes squelettale et des poids (d). Il apparaît que la dernière méthode semble la plus naturelle. La figure 9.15 confirme ce résultat en montrant que l'apport des poids relâche le problème et permet d'éviter des intersections inter-maillage.

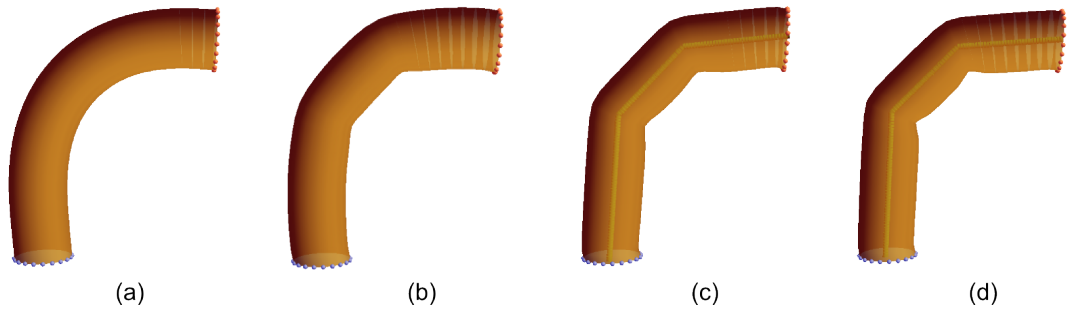


FIGURE 9.14 – Présentation des différentes méthodes de déformation de maillage volumique : ARAP classique (a), en attribuant des poids aux sommets (b), avec une contrainte de rigidité induite par le squelette (c), en attribuant des poids automatiques corrélés à la rigidité induite par le squelette (d).

#### 9.4.6 Déformation finale du maillage de référence

Nous montrons dans cette section comment utiliser la technique des *Green Coordinates* (GC) décrite par Lipman [108] pour déformer le maillage de référence. Soit le maillage de référence  $\mathcal{R} = (X, F)$ , où  $F$  décrit l'ensemble des faces de  $\mathcal{R}$  et  $X$  est l'ensemble des coordonnées Euclidiennes des sommets. Le maillage de contrôle  $\mathcal{V}$  est composé d'un ensemble de tétraèdres, également appelé *cages*, obtenues par la tétraédrisation de Delaunay. Chaque cage  $C$  est définie par  $C = (V_c, F_c)$  où  $V_c$  est l'ensemble des sommets de la cage (4 dans notre cas), et  $F_c$  l'ensemble des faces (4 également). La méthode GC nous donne l'application  $A$  qui calcule  $x'$ , le sommet déformé de  $x$ , appartenant à la cage  $C$  :

$$\mathbf{x}' = A(\mathbf{x}, C') = \sum_{i \in |V|} \phi_i(\mathbf{x}) \mathbf{v}'_i + \sum_{j \in |F|} \psi_j(\mathbf{x}) s_j \mathbf{n}(f'_j) \quad (9.18)$$

où  $C'$  est la cage modifiée de  $C$ . Les "coordonnées"  $\phi$  et  $\psi$  ainsi que le coefficient  $s$  sont calculés "off line" et leur détermination est décrite dans l'article de Lipman [108].

### 9.4.7 Algorithme

Finalement, nous étendons l'algorithme présenté dans la section 6 par l'algorithme 3.

**Données:**

- $X$  : Vecteur des sommets originaux du maillage de référence
- $X'$  : Vecteur des positions du maillage de référence après édition
- $V$  : Vecteur des sommets originaux du maillage de contrôle
- $V'$  : Vecteur des positions du maillage de contrôle après édition
- $U$  : Un ensemble de contraintes de position
- $\mathcal{M}$  : Le maillage de contrôle original

```

1 ComputeGreenCoordinates( $X, \mathcal{M}$ )
2 // Contraintes Laplaciennes et contraintes de position (offline) :
3  $V' = \operatorname{argmin}_V E_l(V) + E_p(V)$ 
4 tant que l'animation est jouée faire
5   tant que pas de convergence faire
6     Trouver  $\Gamma$  à partir de  $\operatorname{argmin}_V E_s(V, \mathcal{M})$  (Équation 6.4), pré-conditionné avec
        $V'$  ;
7     
$$V' = \operatorname{argmin}_V \begin{pmatrix} w_l L \\ w_u U_l \\ w_d D \end{pmatrix} V = \begin{pmatrix} \Delta^A \\ w_u U \\ w_d \Gamma \end{pmatrix}$$

8    $X' = \text{GreenCoordinates}(V', \mathcal{M})$ 

```

**Algorithm 3:** Déformation volumique du maillage avec les contraintes implicites du squelette

## 9.5 Implémentation et résultats

Dans cette section, nous présentons les résultats de notre méthode appliquée à de l'animation de maillages articulés guidés par différents types de données de faible dimension. Notre implémentation est codée en C++ et utilise le solveur de Cholesky pour matrices éparées Cholmod [36] sur un Core2 Duo 2,7 GHz avec 3 Go de RAM et sur Windows XP Pro.

Dans un premier temps nous revenons sur des tests basiques confirmant les objectifs à atteindre. Puis nous appliquons notre méthode à 3 expériences. Les deux premières montrent l'efficacité de notre méthode dans le cadre de problèmes classiques d'édition et d'animation de maillage : tout d'abord en s'appuyant sur un ensemble de points de contrôle édités par l'utilisateur, puis en guidant l'animation à partir des positions des articulations du squelette. Dans

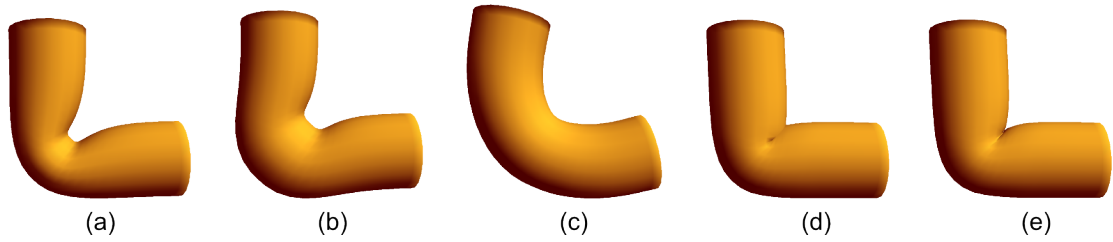


FIGURE 9.15 – Exemple de déformations suivant une rotation de  $90^\circ$  sur un cylindre (622 sommets, 1240 triangles dans le cas surfacique et 878 sommets, 7126 triangles dans le cas volumique) avec différentes techniques : *skinning* classique (a), *skining* utilisant les quaternions duaux (b), *ARAP* volumique (c), *ARAP* volumique avec contraintes de rigidités liées au squelette (d), enfin en intégrant des poids corrélé à l’importance de la déformation (e).

la dernière expérience, nous appliquons notre système à de l’animation de géométrie de haute résolution guidée directement par un ensemble de marqueurs capturés de faible dimension.

### 9.5.1 Tests basiques

**ARAP surfacique vs volumique** La figure 9.8 précédemment introduite compare la technique *ARAP* surfacique classique avec notre méthode *ARAP* volumique pour deux déformations aléatoires. Le modèle surfacique du parallélépipède possède 354 sommets et 1431 triangles contre 363 sommets et 2409 triangles pour le modèle volumique (les temps de calcul d’une itération pour les méthodes surfacique et volumique sont respectivement de 0.008 s et 0.01 s). Nous pouvons remarquer que pour des rotations de forte amplitude, la méthode volumique préserve mieux le volume alors que la méthode surfacique tend à rétrécir la géométrie au niveau des pliures. Le calcul de l’erreur relative par rapport au volume de la géométrie initiale

$$RE - V = \frac{|V_{Original} - V_{Courant}|}{V_{Original}}, \quad (9.19)$$

est de 0.072 pour la méthode surfacique contre 0.018 dans le cas volumique pour la première déformation et de 0.2205 contre 0.1345 dans le cas d’une déformation de type *twist*. Lors de fortes déformations, la méthode volumique préserve également mieux les intersections inter-maillage. Ce résultat s’explique logiquement puisque la méthode *ARAP* préserve la rigidité des cellules à l’intérieur du maillage et donc la cohérence spatiale globale du maillage.

**Comparaisons avec des méthodes classiques** Dans un autre contexte, les figures 9.15 et 9.16 comparent nos résultats avec les méthodes classiques de type simple *skinning* et *skinning* par les quaternions duaux appliqués à une rotation de  $90^\circ$  et à un *twist* de  $360^\circ$ . La première expérience est guidée par le squelette alors que la deuxième est guidée d’une part par le squelette pour les méthodes basées *skinning*, et d’autre part par des points de contrôle pour notre méthode. Pour l’ensemble des tests, nous avons donné comme paramètres :  $w_u = 1.0$ ,  $w_d = 10.0$



(équation 6.4) et  $k = 2$  (équation 9.16). Les temps, différences de volumes et erreurs liées au calcul de la rigidité locale (taille des arêtes après déformation) ainsi que la taille des segments du squelette sont exprimés dans le tableau 9.2. La différence relative de volume  $RE - V$  est décrite par l'équation (9.19), l'erreur correspond à l'énergie exprimant la rigidité globale du système  $RRMS - E$  est exprimée par l'équation (9.6). Enfin l'erreur liée à la préservation des longueurs des segments est donnée par :

$$ED - E = \sum_{(i,j) \in D} \frac{|d'_{ij} - d_{ij}|}{d_{ij}}, \quad (9.20)$$

où  $d'_{ij}$  est la taille de l'arête après déformation. Notre méthode, bien que temps réel, est nettement plus lente que les méthodes classiques d'animation de maillage par *skinning* ou encore en utilisant les quaternions duaux. Toutefois, elle présente l'avantage de mieux préserver le volume global du maillage mais également de rendre un aspect plus naturel à la déformation au niveau des pliures. L'erreur liée au bon respect de la taille des segments du squelette est logiquement nulle du fait que l'animation même du squelette impose le respect de cette contrainte. La figure 9.15 (e) montre bien visuellement l'intérêt d'introduire de la rigidité par des contraintes liées au squelette. Nous observons également que le relâchement de la préservation de la rigidité locale au niveau des pliures permet d'éviter des collisions inter-maillage et d'améliorer l'aspect naturel de la déformation.

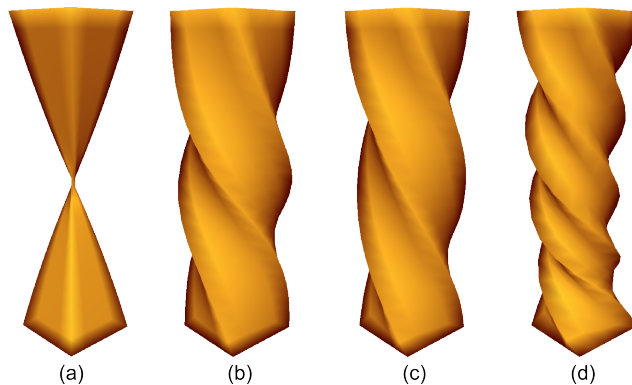


FIGURE 9.16 – Déformation de type *twist* d'un parallélépipède de  $360^\circ$  (354 sommets, 1431 triangles dans le cas surfacique et 363 sommets, 2409 triangles dans le cas volumique) : technique de *skinning* (a), quaternions duaux (b), notre méthode (c), notre méthode appliqué à  $720^\circ$  (d).

### 9.5.2 Édition de maillage guidée par des sommets définis manuellement

Dans cette première expérience, nous voulons mettre en évidence la capacité de notre système à résoudre des problèmes d'édition de maillage classique. L'application doit par ailleurs

Modèle	Temps	RE-V	RRMS-E	ED-E
Fig 9.15 (a)	0.0003	0.056		
Fig 9.15 (b)	0.0002	0.037		
<b>Fig 9.15 (c)</b>	<b>0.0305</b>	<b>0.018</b>	<b>0.084</b>	
<b>Fig 9.15 (d)</b>	<b>0.0325</b>	<b>0.018</b>	<b>0.066</b>	<b>0</b>
<b>Fig 9.15 (e)</b>	<b>0.0345</b>	<b>0.023</b>	<b>0.076</b>	<b>0</b>
Fig 9.16 (a)	0.0001	0.6		
Fig 9.16 (b)	0.0001	0.074		
<b>Fig 9.16 (c)</b>	<b>0.0125</b>	<b>0.010</b>	<b>0.169</b>	<b>0</b>
Fig 9.19 (a)	0.0001	0.042		
<b>Fig 9.19 (b)</b>	<b>0.2188</b>	<b>0.034</b>	<b>0.056</b>	<b>0.0005</b>

TABLE 9.2 – Temps, volume et erreurs des figures 9.15, 9.16 et 9.19.  $RE - V$  est décrite par l'équation (9.19),  $RRMS - E$  correspond à l'erreur exprimée par l'équation (9.6) et  $ED - E$  à l'erreur correspondant à l'équation (9.20). En gras les résultats obtenus par notre méthode.



FIGURE 9.17 – Déformation d'Armadillo. A gauche la pose de référence, les autres poses sont des déformations aléatoires.

préservé le volume de la géométrie, respecter les propriétés du squelette, et utiliser la technique de cage pour améliorer le temps de calcul. En fixant plusieurs sommets du maillage de contrôle et en éditant les autres, nous montrons tout d'abord que notre méthode est capable d'éditer interactivement un maillage de haute résolution. Les résultats sont illustrés par les figures 9.17 et 9.18. Par ailleurs, le tableau 9.3 montre que notre méthode est temps réel tant que le maillage de contrôle n'est pas trop détaillé. Les mesures de volume  $RE - V$  montrent que les déformations obtenues ont des volumes proches et que la préservation de cette caractéristique est directement liée à la qualité du maillage de contrôle (plus le maillage de contrôle est détaillé, plus le volume est préservé). C'est également le cas pour la préservation des propriétés de la géométrie qui sont globalement bien préservées puisque l'erreur liée à l'énergie de déformation  $RRMS - E$  donnée par l'équation 9.9 donne de faibles valeurs. Enfin à partir des résultats donnés par  $ED - E$ , nous informons que les propriétés de rigidités induites par le squelette sont bien respectées puisque l'erreur relative correspondant aux longueurs des segments

Modèle (pourcentage d'information du maillage de contrôle comparé au maillage référence)	nombre de sommets et triangles du maillage de référence	nombre de sommets et triangles du maillage de contrôle	Temps de construction du système de Cholesky (offline)(sec.)	Temps de mise à jour du maillage de contrôle (sec.)	Temps de mise à jour du système GC (sec.)	RRMS-E	RE-V du maillage de référence	Erreur de longueur des arêtes du squelette : ED-E
figure 9.17 (b) (0.24%)	(172914, 345944)	(423,5092)	0.101	0.012	0.063	0.04	0.042	0.19
figure 9.17 (b) (15.81%)	(172914, 345944)	(27350,36814)	264.432	0.986	0.236	0.027	0.029	0.011
figure 9.18 (b) (100%)	(19851, 39698)	(25429,200375)	310.782	1.013	0	0.066	0.034	0
figure 9.23 (a) (7.32%)	(5023, 10042)	(368, 4728)	0.102	0.012	0.002	0.087	0.061	0.016
figure 9.21 (b) (23.57%)	(8256, 15198)	(1946, 26364)	1.836	0.068	0.014	0.101	0.012	0.019

TABLE 9.3 – Tableau des temps et erreurs pour différents maillages de référence et de contrôle après 50 itérations.

avant et après déformation est quasiment nulle.



FIGURE 9.18 – Déformation volumique du modèle *horse*. A gauche la pose de référence, les autres poses sont des déformations aléatoires.

### 9.5.3 Animation de maillage guidée par le squelette

Nous appliquons ensuite notre méthode à de l'animation de maillage guidée par l'animation d'un squelette issue de la capture de mouvement. Dans ce but, nous donnons en entrée les positions des articulations. Il est intéressant de noter que notre système n'a pas besoin d'étape de *rigging* et ne nécessite pas de T-pose commune entre le maillage et le squelette animé.

**Animation de maillage** Notre première expérimentation est dédiée à la validation de notre méthode sur l'animation simple d'un maillage guidé par un squelette aux mêmes dimensions. Nous avons testé notre système sur l'animation d'un humanoïde et comparé les résultats avec de l'animation par *skinning* classique. Le modèle est constitué de 605 sommets et 3206 triangles

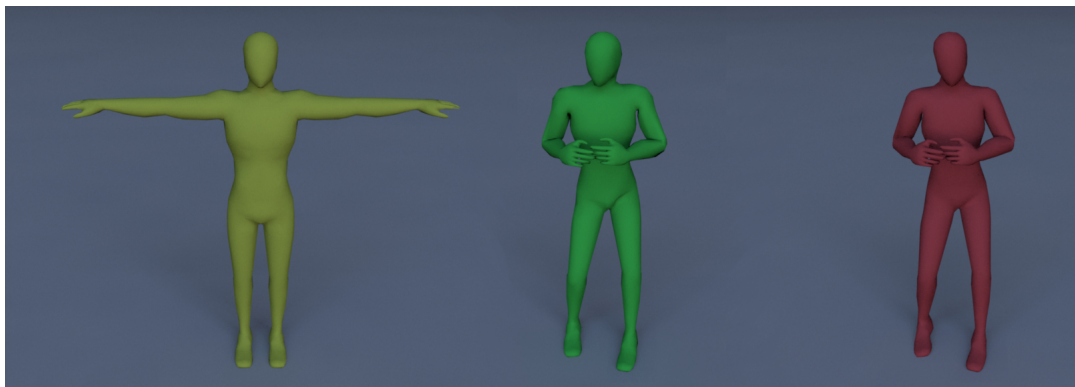


FIGURE 9.19 – Comparaison entre l’animation de maillage par skinning classique (en vert) et par notre méthode (en rouge). En jaune la T-pose.

pour le maillage surfacique et 6395 sommets et 52135 triangles pour le maillage volumique. Il apparait en observant la figure 9.19 que notre méthode fournit des déformations plus naturelles au niveau des épaules avec une meilleure préservation locale du volume au niveau des aisselles. Cette impression est confirmée par les valeurs obtenues dans le tableau 9.2. On note toutefois que notre méthode étant guidée uniquement par les positions, le système est sous contraint. Nous observons bien les inconvénients d’un tel contrôle puisque nous perdons l’information de rotation le long des axes des segments (orientation de la tête dans la figure 9.19). Néanmoins, cette technique est une bonne alternative à l’animation de maillage classique.

**Transfert d’animation** Comme le montre la vidéo accompagnant ce document, notre système permet un transfert facile de l’animation d’un maillage vers un autre, simplement par l’inclusion des positions des articulations dans le maillage de contrôle. La vidéo montre un exemple d’animation de plusieurs maillages de contrôle avec un squelette unique. Nous pouvons par ailleurs voir qu’un faible redimensionnement est acceptable avec un tel procédé. Toutefois notre méthode ne prend pas en compte l’ensemble des contraintes du maillage. Par exemple, certaines intersections apparaissent dans le cadre du transfert d’animation d’un humain mince vers un plus enveloppé.

**Application à notre méthode d’édition de mouvement 3D+t** Nous mettons également en évidence par la figure 9.20 que notre système répond à une des problématiques levée dans la section 7. En effet, notre système travaillant dans l’espace Laplacien, nous avons obtenu à l’issue de la minimisation uniquement les positions des articulations comme information de contrôle pour le maillage, contrôle permettant d’appliquer notre méthode développée dans cette section. Ainsi, la combinaison des deux méthodes permet de : (i) transférer efficacement l’animation d’un maillage vers un autre avec un fort redimensionnement ; (ii) de préserver certaines relations spatiales comme le contact ou les intersections inter-maillage évoquées précédem-

ment.



FIGURE 9.20 – Exemple de trois cas d’adaptation morphologique de mouvement. Le squelette en jaune est adapté à trois morphologies fortement différentes.

#### 9.5.4 Animation de maillage guidée par les marqueurs

L’objectif principal de ce travail est de proposer un système produisant des animations de maillage haute résolution guidées directement par des données issues de la capture de mouvement. Dans ce contexte, nous montrons à travers divers exemples que notre méthode est efficace pour produire des animations de haute qualité en temps réel.



FIGURE 9.21 – Animation d’un maillage quelconque. A gauche, première figure : la géométrie de référence et le positionnement manuel des marqueurs. Après : cinq postures générées par notre système.

Dans le premier exemple, nous appliquons notre méthode à l’animation d’un maillage composé de 8256 sommets et 15198 triangles, guidé par seulement 36 marqueurs (huit marqueurs pour chaque main, deux pour les pieds, quatre pour la tête et douze pour le reste du corps). Les marqueurs sont placés à l’intérieur et à l’extérieur du maillage de contrôle en fonction de la différence de morphologie entre le modèle capturé et la géométrie cible. Comme le montre la figure 9.21 et dans la vidéo jointe à ce manuscrit, le mouvement des marqueurs est suivi avec succès et les déformations résultantes ont un aspect naturel. Pour une évaluation quantitative de notre algorithme, le tableau 9.3 montre que les contraintes sont bien préservées et la

figure 9.22 montre la bonne convergence de notre algorithme de descente alternée qui minimise simultanément les énergies de déformation, tout en respectant les propriétés du squelette.

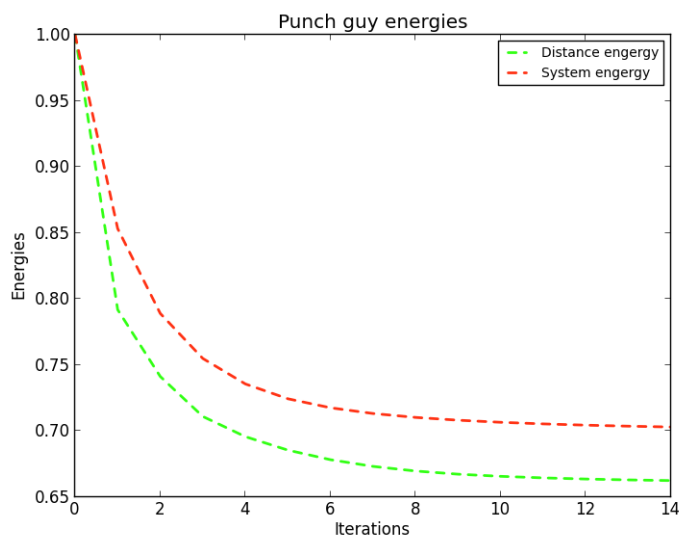


FIGURE 9.22 – Exemple de convergence de notre algorithme de descente alterné pour un intervalle de 10 *frames* durant l’animation. Les énergies de distance et du système correspondent aux valeurs normalisées décrite dans le tableau 9.3

Dans le deuxième exemple, nous animons les maillages de deux modèles de main pour un mouvement en langue des signes française. Animer des mains dans le cadre de LSF constitue un défi : cela nécessite une grande précision, en particulier pour produire des configurations de mains plausibles. Par ailleurs, de faibles modifications sur certaines postures ou sur les trajectoires peuvent modifier la signification des signes produits [56]. Dans notre expérience, chaque main est représentée par une géométrie de haute résolution (5023 sommets pour 10042 triangles) et entraînée par quelques points de contrôle (huit marqueurs par main).

Dans la vidéo qui accompagne ce document, nous comparons notre méthode avec une animation basée sur du *skinning* classique. Dans les deux cas, nous partons des mouvements originaux des mains, le mouvement des marqueurs capturés étant représenté par des signaux de faible dimension et de faible qualité. Comme nous l’avons déjà mentionné dans ce manuscrit, il est difficile de positionner un grand nombre de marqueurs sur les mains. De plus, la langue des signes conduit à de nombreuses occultations des trajectoires des marqueurs nécessitant un post traitement de reconstruction. Dans le cadre de l’animation par *skinning*, la vidéo met en évidence les difficultés rencontrées pour reconstruire l’animation du squelette pour un modèle de dimension différente, notamment par les interpénétrations du maillage et de certains écarts au niveau des contacts. Elle permet également de mettre en évidence les limites de l’étape de *rigging* spécifique à la main dues à l’attribution de poids imprécis et d’une structure du

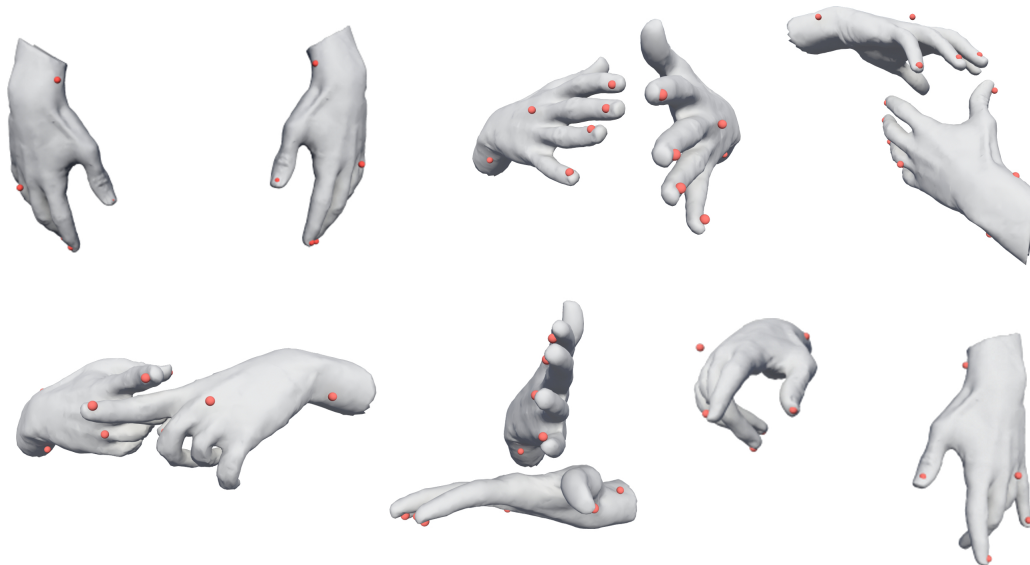


FIGURE 9.23 – Animation des mains appliquée à la langue des signes : six postures générées par notre système. L’animation des marqueurs (en rouge) guide les déformations du maillage.

squelette approximative.

Au contraire, comme illustré par la figure 9.23 et la vidéo jointe, notre méthode anime avec succès les deux mains : les animations sont réalistes. Les maillages sont soumis à des déformations importantes mais comme présenté dans le tableau 9.3, les animations préservent les détails géométriques et les propriétés du squelette. Nous pouvons par ailleurs observer sur des séquences animées les résultats de notre méthode pour différents cas : avec un squelette sans limite angulaire, et avec un squelette couplé à des limites angulaires (conférer la vidéo accompagnant ce manuscrit). Enfin, les résultats sont convaincants et démontrent l’intérêt d’un tel *pipeline* pour ce type d’animation.

## 9.6 Conclusion et perspectives

Nous avons présenté dans ce chapitre une technique d’animation et d’édition de maillage interactive qui couple un maillage volumique avec la structure du squelette sous-jacent. L’originalité de la méthode réside dans ce couplage qui nous permet d’intégrer les contraintes du squelette implicite dans le processus de déformation. Le concept de rigidité est amélioré par plusieurs aspects. En étendant le paradigme *as-rigid-as-possible* aux maillages volumiques, nous évitons des déformations non naturelles et préservons mieux le volume global de l’objet. Lors de la construction du maillage volumique, nous incluons une structure interne résultant d’une discrétisation spatiale du squelette. Cette structure joue un rôle central dans le processus de tétraédrisation. De plus, les propriétés de rigidité du système sont améliorées en tenant compte des caractéristiques du squelette telles que la longueur des segments et les limites arti-

culaires. Comme les contraintes liées au squelette ne sont pas linéaires, notre système alterne par une méthode de minimisation alternée, une déformation Laplacienne linéaire du maillage volumique et le calcul des postures respectant ces contraintes. Afin de produire des postures naturelles, nous pensons que l'inclusion de ces contraintes est primordiale pour l'animation de corps articulés, comme en attestent les animations des mains en LSF. Enfin, notre processus d'animation est stable et robuste, et nous permet de produire des animations de maillage de haute qualité en temps réel à partir d'un grand nombre de classes de mouvements capturés, même si la dimension des marqueurs capturés est fortement réduite.

L'étude approfondie du couplage du maillage/squelette ouvre des perspectives prometteuses pour l'animation de personnages articulés. Une extension peut être envisagée sous la forme d'une discrétisation spatiale non uniforme du squelette, ce qui permettrait de mieux préserver localement certaines propriétés lors de grandes déformations. Plus généralement, il serait intéressant d'étudier l'influence du couplage sur la convergence de la méthode. Dans un futur proche, nous prévoyons d'explorer la possibilité d'extraire automatiquement les maillages *cage*, ce qui est essentiel pour obtenir de bons résultats et faciliter le travail de l'utilisateur. Enfin, nous pensons déporter l'essentiel des calculs sur le GPU pour accélérer significativement la méthode comme l'a fait récemment Zollhofer [196].





# 10

## Conclusion

### Sommaire

---

<b>10.1 Contributions et discussions</b> . . . . .	<b>147</b>
<b>10.2 Perspectives</b> . . . . .	<b>149</b>

---

### 10.1 Contributions et discussions

L'objectif de ce travail de thèse était de proposer de nouvelles représentation du mouvement par le biais d'un ensemble de relations spatiales, ceci dans le but de corriger, modifier ou adapter le mouvement capturé à de nouveaux contextes. A travers cette démarche l'objectif était également de proposer un processus alternatif d'animation de maillage guidée par des données capturées permettant d'améliorer la qualité de l'animation finale.

Pour répondre à ces questions, nous avons proposé deux axes d'études : tout d'abord en caractérisant le mouvement par un ensemble de distances, puis en caractérisant sa géométrie par des informations différentielles.

Dans un premier temps, en nous plaçant dans l'espace métrique, nous avons proposé de re-définir les postures d'un mouvement basé squelette par un ensemble de distances entre les articulations, afin d'encoder les relations spatiales de chaque articulation avec le reste du squelette et de l'environnement. La reconstruction d'une posture (ensemble des positions des articulations) se fait au moyen d'une descente de gradient qui privilégie la préservation des longueurs des segments tout en conservant au mieux les autres contraintes de distance. A travers plusieurs applications, nous avons montré qu'il est possible de résoudre des problèmes classiques de cinématique directe ou inverse.

L'intérêt majeur de notre méthode réside dans une caractérisation du contrôle du mouvement plus simple et plus intuitif qu'avec les techniques classiques : la génération ou l'édition du mouvement s'obtient par un ensemble de contraintes de distance à minimiser plutôt que des configurations spatiales à obtenir. Nous avons également vu qu'en caractérisant une articulation

vis-à-vis de l'ensemble ou d'une partie du squelette, il était possible de préserver les relations spatiales entre les articulations pour la génération de mouvement de structures articulées complexes. Par ailleurs, contrairement à la plupart des méthodes qui s'appuient sur une structure arborescente, notre contribution permet de travailler sur des graphes avec cycles. Dans un autre contexte, appliqué à l'analyse de mouvement, la redondance induite par la représentation du mouvement présente un intérêt pour caractériser et réduire la dimension du mouvement. Nous l'avons en particulier observé dans le cadre d'une application originale de recherche de mouvements dans des grandes bases de données.

Cependant l'approche que nous avons développée présente un certain nombre de limitations. Rappelons tout d'abord que les informations qui caractérisent le mouvement sont redondantes. D'une part, cela sur-construit le problème, ce qui peut amener dans certaines configurations à des minima locaux, c'est en particulier le cas pour une édition du mouvement avec des objectifs à atteindre trop éloignés. D'autre part une telle représentation s'applique bien sur des graphes de petite taille mais semble difficilement exploitable sur des graphes plus denses, la complexité du problème croissant en fonction du nombre d'articulations considérées. Enfin, cette approche traite de l'information temporelle de manière implicite, et n'assure donc pas la préservation de la dynamique du mouvement.

Dans un second temps, nous avons défini le mouvement par une séquence de graphes dont les sommets sont caractérisés par une information différentielle. A travers cette représentation, nous avons proposé une méthode originale et robuste permettant d'éditer un mouvement en préservant de manière optimale les relations spatiales entre les sommets. Ces relations spatiales sont de deux types : (i) l'expression Laplacienne des sommets ; (ii) les distances entre plusieurs sommets constituant des contraintes "inviolables". Pour ce faire, nous avons conçu une méthode originale couplant une partie linéaire dédiée à la préservation des coordonnées différentielles Laplaciennes et une autre non linéaire permettant de préserver certaines distances entre les sommets du graphe.

Notre approche a donné lieu à plusieurs applications permettant de résoudre au moyen de l'adaptation du modèle précédent des problèmes classiques rencontrés en animation par ordinateur :

- nous avons proposé (chapitre 7) une représentation du mouvement encodant d'une manière unifiée l'information spatiale et temporelle d'un mouvement. Cette représentation permet en particulier d'introduire une dépendance temporelle entre plusieurs poses successives, ce qui n'est pas le cas des méthodes géométriques classiques. Cette propriété permet ainsi de préserver la dynamique du mouvement. La nouvelle méthode développée peut être comparée à celle considérant le mouvement dans l'espace métrique (partie 3) : elle découple les priorités entre les contraintes en deux systèmes de minimisation, linéaire et non linéaire, ce qui permet de réduire la complexité du problème. La majeure partie des contraintes étant minimisée au moyen d'un système linéaire, cette nouvelle approche évite les minima locaux et est donc plus robuste.

Cette méthode répond à un éventail assez large de problématiques telles que la modification spatiale de certaines trajectoires, le redimensionnement du squelette tout en préservant la signification du mouvement. Il est à noter également qu'elle s'applique à tout type de mouvement. Enfin nous avons vu que la caractérisation par des relations spatiales permet de préserver des informations de proximité telles que le contact entre deux effecteurs lors de l'édition ;

- dans le cadre de la reconstruction de trajectoires de marqueurs issus de données capturées, nous avons formulé l'hypothèse qu'il existait une corrélation spatiale et temporelle entre le déplacement d'un marqueur et celui de ses voisins. Cela nous a permis d'utiliser notre méthode d'édition de graphe au moyen d'un algorithme *forward-backward*. Dans ce contexte applicatif, les informations différentielles permettent de préserver les relations spatiales entre les marqueurs alors que les contraintes de distance permettent de personnaliser la reconstruction des trajectoires ;

- nous avons dans une dernière contribution proposé un nouveau processus d'animation de personnage directement guidée par les marqueurs issus de données capturées. Ainsi, nous avons appliqué notre méthode d'édition de graphe à une représentation du mouvement par des maillages 3D. Dans ce contexte, contrairement aux processus classiques qui guident le maillage par l'animation du squelette, notre méthode déforme le maillage contraint par des propriétés inhérentes au squelette. Cette nouvelle contribution permet de limiter le nombre de post-traitements du processus classique et en particulier la phase de *rigging*. Nous avons rejoué avec succès des séquences animées mettant en jeu des structures articulées complexes telles que des mains : les animations reproduisent correctement les postures de contact, ce qui constitue une amélioration significative de la qualité des données vis-à-vis des résultats obtenus par le processus classique.

Les différents modèles développés répondent au problème initialement posé : il est ainsi possible de corriger des trajectoires de données capturées entachées d'erreurs induites par la chaîne d'acquisition et de post-traitements. Notre approche permet d'éditer les données du mouvement, soit en manipulant les squelettes ou les maillages 3D, tout en tenant compte des contraintes liées à l'expressivité des gestes. Nous avons par exemple appliqué avec succès notre méthode à l'animation des mains d'une signeuse virtuelle.

## 10.2 Perspectives

Cette thèse donne lieu à plusieurs perspectives. Nous détaillons ci-après celles qui sont directement liées aux concepts développés dans ce document ; puis dans un second temps, nous présentons certaines perspectives à plus long terme.

### Perspectives directes

Notre premier axe d'étude a conduit à proposer une nouvelle représentation du mouvement dans l'espace métrique qui est particulièrement intéressante, en particulier en ce qui concerne

l'analyse de mouvement et la recherche de mouvement dans de grandes bases de données. Rappelons qu'elle est invariante aux transformations rigides (rotation et translation), et permet également de s'abstraire de la topologie de la structure de données considérée. Ces deux caractéristiques présentent un intérêt certain pour l'analyse et la caractérisation de tout type de structure de données ; mouvement ou autres. Nous pensons par exemple que cette représentation pourrait s'appliquer à la recherche en temps réel de maillages (entiers ou partiels) dans des bases de données.

Notre second axe d'étude concernant l'édition de graphe caractérisé par une information différentielle, nous a permis de répondre plus largement aux objectifs fixés dans cette thèse. La méthode proposée pourrait facilement s'appliquer à d'autres contextes et ainsi offrir d'autres perspectives, comme détaillé ci-après :

En ce qui concerne l'édition de mouvement utilisant l'opérateur Laplacien  $3D+t$ , plusieurs perspectives sont envisagées. Parmi celles-ci, il serait intéressant d'améliorer le système en introduisant des contraintes physiques ou des contraintes angulaires, ceci afin de rendre les résultats plus réalistes. Une autre possibilité intéressante serait d'étendre le principe d'utilisation d'*interaction mesh*<sup>1</sup> proposé par Ho et Komura [78] afin d'introduire automatiquement de nouvelles relations entre la scène et les différentes articulations des personnages. Cela permettrait de proposer des opérations d'adaptation de mouvement plus complexes semblables à celles proposées par l'auteur, tout en utilisant l'opérateur  $3D+t$ . Enfin, en s'inspirant des travaux appliqués à l'interpolation de structures géométriques [2, 156, 167], nous pensons que l'opérateur Laplacien pourrait également être un outil intéressant pour générer des transitions entre les mouvements.

Notre dernière contribution propose un processus alternatif d'animation de maillage guidé par des données capturées. Remarquons qu'à travers cette étude, nous établissons le lien entre deux domaines de recherche de l'informatique graphique : l'animation de systèmes articulés et la déformation de maillage. A l'image de certains travaux récents, nous pensons que cette voie est à développer, en appliquant par exemple notre approche à des techniques récentes plus souples qu'*ARAP* (comme celle proposée par Jacobson [85]). Par ailleurs, cette application est prometteuse pour plusieurs raisons : à condition de proposer une méthode permettant d'automatiser la création du maillage cage (par exemple celle proposée par Xian [184]), le processus d'animation du maillage directement guidée par les données capturées est vraiment simple et rapide à mettre en oeuvre. De plus, le rapport entre le temps passé par un animateur sur les différents traitements et la qualité du résultat peut être bien plus optimisé en utilisant notre méthode plutôt qu'en suivant le processus classique. Enfin, la représentation du maillage à travers son information différentielle permet d'appliquer facilement des méthodes de déformations

---

1. Dans son article, Ho propose de créer un maillage pour représenter une scène : le *maillage d'interaction*. Ce maillage volumétrique encode les relations spatiales entre les éléments caractérisant la scène (les articulations de un ou plusieurs squelettes avec certains éléments de la scène). Il est créé par une tétraédrisation de Delaunay.

complémentaires comme les plis ou celles liées aux muscles.

### A plus long terme

Cette étude s'applique aux mouvements corporels complexes tels que le mouvement des mains, les contacts (inter-squelette ou inter-maillage) ou les interactions entre personnages et environnement. Pour illustrer notre propos, nous avons utilisé des configurations propres à la langue des signes. Ainsi nous pensons que nos travaux sont exploitables d'une manière générale dans le cadre de mouvements expressifs. Nous pourrions appliquer nos méthodes et les évaluer leurs résultats sur plusieurs types de gestes expressifs comme par exemple les gestes musicaux, ceux d'acteurs de théâtre ou les gestes de danse. Par ailleurs, il serait intéressant de mener des études perceptives, en particulier concernant de la langue des signes pour évaluer si notre approche permet de mieux préserver la signification des gestes.

Nous avons mis en évidence l'intérêt d'utiliser les relations spatiales afin de préserver certaines informations spécifiques importantes du mouvement. Il s'agit par exemple du contact entre deux doigts. Dans cette thèse, nous considérons ce problème d'un point de vue global, à l'échelle du système poly-articulé. Nous pensons qu'il serait intéressant dans des travaux futurs de prendre davantage en considération les interactions inter-systèmes "locales". Appliqué à de l'animation de maillage, cette idée se traduirait par la préservation de certaines propriétés physiques telles que le *Zero Moment Point*, ou encore la détection des collisions inter-squelette associées aux effets cinématiques. Dans le cadre de l'animation de maillage, il s'agit par exemple de mettre en évidence les déformations locales qu'impliquent le contact entre deux parties du corps, comme l'écrasement ou les plis formés par la peau.

Enfin, nous pensons que certaines de nos applications présentent un intérêt fort dans un cadre industriel. A titre d'exemple, la reconstruction de trajectoires proposée pourrait facilement s'intégrer dans la chaîne de post-traitements des données capturées de logiciels classiques. Cette étape nécessiterait toutefois une validation expérimentale plus exhaustive. Concernant l'application traitant de l'animation de maillage, le milieu industriel s'intéressant à la capture de mouvement a mis en évidence les difficultés de restituer l'animation des parties complexes du corps telles que les mains (*e.g* [65]). Cela semble lié aux limites actuelles des systèmes de capture de mouvement, combinées à celles du processus classique d'animation de maillage. Dans ce contexte, nous sommes convaincus que cette dernière contribution, offre une alternative intéressante au processus classique, en particulier pour l'animation des mains.



# Bibliographie

- [1] Qualisys AB. Qualisys motion capture system, 1989.
- [2] M. Alexa. Local control for mesh morphing. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 209–215, 2001.
- [3] Marc Alexa. Differential coordinates for local mesh morphing and deformation. *The Visual Computer*, 19 :105–114, 2003. 10.1007/s00371-002-0180-0106.
- [4] Marc Alexa and Wolfgang Müller 0004. Representing animations by principal components. *Comput. Graph. Forum*, 19(3) :411–418, 2000.
- [5] Marc Alexa, Daniel Cohen-Or, and David Levin. As-rigid-as-possible shape interpolation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [6] Andreas Aristidou, Jonathan Cameron, and Joan Lasenby. Predicting missing markers to drive real-time centre of rotation estimation. In FranciscoJ. Perales and RobertB. Fisher, editors, *Articulated Motion and Deformable Objects*, volume 5098 of *Lecture Notes in Computer Science*, pages 238–247. Springer Berlin Heidelberg, 2008.
- [7] Andreas Aristidou and Joan Lasenby. Fabrik : A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5) :243 – 260, 2011.
- [8] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(5) :698–700, May 1987.
- [9] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM Trans. Graph.*, 27(3), 2008.
- [10] Matthieu Aubry, Frédéric Julliard, and Sylvie Gibet. Modeling joint synergies to synthesize realistic movements. In *Gesture Workshop*, pages 231–242, 2009.
- [11] Charly Awad, Nicolas Courty, Kyle Duarte, Thibaut Naour, and Sylvie Gibet. A combined semantic and motion capture database for real-time sign language synthesis. In *Proceedings of the 9th International Conference on Intelligent Virtual Agents, IVA '09*, pages 432–438, Berlin, Heidelberg, 2009. Springer-Verlag.
- [12] P. Baerlocher and R. Boulic. An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. *The Visual Computer*, 20(6) :402–417, 2004.
- [13] Paolo Baerlocher and Ronan Boulic. Parametrization and range of motion of the ball-and-socket joint. In *Deformable avatars*, pages 180–190. Springer, 2001.
- [14] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.
- [15] Alexander Belyaev, Hans-Peter Seidel, and Shin Yoshizawa. Skeleton-driven laplacian mesh deformations. Research Report MPI-I-2006-4-005, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, November 2006.



- [16] Michael W. Berry. Large scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6 :13–49, 1992.
- [17] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Trans. Sen. Netw.*, 2(2) :188–220, May 2006.
- [18] Åke Björck. *Numerical methods for least squares problems*. Siam, 1996.
- [19] Jonathan Blow. Inverse kinematics with quaternion joint limits. 2002.
- [20] Leonard M Blumenthal. Distance geometry. In *A study of the development of abstract metrics. With an introduction by Karl Menger, Univ. of Missouri Studies*, volume 13, 1953.
- [21] Péter Borosán, Reid Howard, Shaoting Zhang, and Andrew Nealen. Hybrid mesh editing.
- [22] Mario Botsch and Leif Kobbelt. An intuitive framework for real-time freeform modeling. In *ACM SIGGRAPH 2004 Papers, SIGGRAPH '04*, pages 630–634, New York, NY, USA, 2004. ACM.
- [23] Alexandre Bouenard, Marcelo M. M. Wanderley, and Sylvie Gibet. Gesture control of sound synthesis : Analysis and classification of percussion gestures. *Acta Acustica united with Acustica*, 96(4) :668–677, 2010.
- [24] R. Boulic, R. Mas, and D. Thalmann. A robust approach for the control of the center of mass with inverse kinetics. *Computers & Graphics*, 20(5), 1996.
- [25] Robert Bridson, Joseph Teran, Neil Molino, and Ronald Fedkiw. Adaptive physics based tetrahedral mesh generation using level sets. *Eng. Comput. (Lond)*, 21 :2–18, 2005.
- [26] C. Budd and A. Hilton. Skeleton driven laplacian volumetric deformation. In *Proceedings of the 2009 Conference for Visual Media Production, CVMP '09*, pages 61–68. IEEE Computer Society, 2009.
- [27] J.-F. Cai, E. J. Candes, and Z. Shen. A Singular Value Thresholding Algorithm for Matrix Completion. *ArXiv e-prints*, October 2008.
- [28] B. Le Callennec and R. Boulic. Interactive motion deformation with prioritized constraints. *Graphical Models*, 68(2) :175–193, 2006.
- [29] Benoît Le Callennec and Ronan Boulic. Interactive motion deformation with prioritized constraints. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '04*, pages 163–171, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [30] Steve Capell, Seth Green, Brian Curless, Tom Duchamp, and Zoran Popovic. Interactive skeleton-driven dynamic deformations. *ACM Trans. Graph.*, 21(3) :586–593, July 2002.

- [31] S. Carvalho, R. Boulic, and D. Thalmann. Interactive Low-Dimensional Human Motion Synthesis by Combining Motion Models and PIK. *Computer Animation & Virtual Worlds*, 18, 2007. Special Issue of Computer Animation and Social Agents (CASA2007), to appear.
- [32] P. Cerveri, E. Momi, N. Lopomo, G. Baud-Bovy, R.M.L. Barros, and G. Ferrigno. Finger kinematic modeling and real-time hand motion estimation. *Annals of Biomedical Engineering*, 35(11) :1989–2002, 2007.
- [33] J. Chai and J.K. Hodgins. Constraint-based motion optimization using a statistical dynamic model. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 26(3), July 2007.
- [34] Jinxiang Chai and Jessica K. Hodgins. Performance animation from low-dimensional control signals. *ACM Trans. Graph.*, 24(3) :686–696, July 2005.
- [35] Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. A simple geometric model for elastic deformations. *ACM SIGGRAPH 2010 papers on - SIGGRAPH '10*, page 1, 2010.
- [36] Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. Algorithm 887 : Cholmod, supernodal sparse cholesky factorization and update/down-date. *ACM Trans. Math. Softw.*, 35(3) :22 :1–22 :14, October 2008.
- [37] H-Y. Chiu, F.C. Su, S-T. Wang, and H-Y. Hsu. The motion analysis system and goniometry of the finger joints. *The Journal of Hand Surgery : British and European Volume*, 23(6) :788 – 791, 1998.
- [38] K.J. Choi and H.S. Ko. Online motion retargetting. *Journal of Visualization and Computer Animation*, 11 :223–235, 2000.
- [39] David I Chu, Hunter C Brown, and Moody T Chu. On least squares euclidean distance matrix approximation and completion. *Available at February*, 16, 2010.
- [40] R.R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1) :5–30, 2006.
- [41] N. Courty and E. Arnaud. Inverse kinematics using sequential Monte Carlo methods. In *International Conference on Articulated Motion and Deformable Object*, pages 1–11, Port d’andratx, Espagne, 2008.
- [42] N. Courty, E. Marchand, and B. Arnaldi. Through-the-eyes control of a virtual humanoid. In *Proc. of Computer Animation 2001*, pages 74–83, Seoul, South Korea, November 2001.
- [43] Nicolas Courty and Anne Cuzol. Conditional Stochastic Simulation for Character Animation. *Computer Animation and Virtual Worlds*, pages 1–10, 2010.
- [44] Nello Cristianini, Jaz Kandola, Andre Elisseeff, and John Shawe-Taylor. On kernel target alignment. In DawnE. Holmes and LakhmiC. Jain, editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 205–256. Springer Berlin Heidelberg, 2006.

- [45] B. Cutler, J. Dorsey, and L. McMillan. Simplification and improvement of tetrahedral models for simulation. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, SGP '04*, pages 93–102, New York, NY, USA, 2004. ACM.
- [46] Robert B Davies. Newmat c++ matrix library. [http://www.robertnz.net/nm\\_intro.htm](http://www.robertnz.net/nm_intro.htm), 2006.
- [47] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99*, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [48] Tamal K. Dey, Pawas Ranjan, and Yusu Wang. Eigen deformation of 3d models. *Vis. Comput.*, 28(6-8) :585–595, June 2012.
- [49] Q. Dong and Z. Wu. A geometric build-up algorithm for solving the molecular distance geometry problem with sparse distance data. *Journal of Global Optimization*, 26 :321–333, July 2003.
- [50] Klaus Dorfmüller-Ulhaas. Robust optical user motion tracking using a kalman filter. In *10th ACM Symposium on Virtual Reality Software and Technology*. Citeseer, 2003.
- [51] A. Dubrovina and R. Kimmel. Matching shapes by eigendecomposition of the Laplace-Beltrami operator. In *3DPVT*, 2010.
- [52] Jack D. Dunitz. Distance geometry and molecular conformation. *Journal of Computational Chemistry*, 11(2) :265–266, 1990.
- [53] Haw-ren Fang and Dianne P O’Leary. Euclidean distance matrix completion problems. *Optimization Methods and Software*, 27(4-5) :695–717, 2012.
- [54] Noura Faraj, Jean-Marc Thiery, Isabelle Bloch, Nadège Varsier, Joe Wiart, and Tamy Boubekeur. Robust and scalable interactive freeform modeling of high definition medical images. *Mesh Processing in Medical Image Analysis 2012*, pages 1–11, 2012.
- [55] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2) :298–305, 1973.
- [56] S Gibet, N Courty, K Duarte, and T L Naour. The signcom system for data-driven animation of interactive virtual signers : Methodology and evaluation, 2011.
- [57] S. Gibet and P.F. Marteau. A self-organised model for the control, planning and learning of nonlinear multivariable systems using a sensori- feedback. *Journal of Applied Intelligence*, 4 :337–349, 1994.
- [58] Sylvie Gibet, Nicolas Courty, Kyle Duarte, and Thibaut Le Naour. The signcom system for data-driven animation of interactive virtual signers : Methodology and evaluation. *ACM Trans. Interact. Intell. Syst.*, 1(1) :6 :1–6 :23, October 2011.

- [59] M. Gleicher. Retargetting motion to new characters. In *Proc. of the 25th annual conference on Computer graphics and interactive techniques (Siggraph)*, pages 33–42, 1998.
- [60] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques, SIGGRAPH '98*, pages 33–42. ACM, 1998.
- [61] W. Glunt, T. L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *J. Comput. Chem.*, 14(1) :114–120, January 1993.
- [62] Eitan Grinspun, Mathieu Desbrun, Konrad Polthier, Peter Schröder, and Ari Stern. Discrete differential geometry : an applied introduction. *ACM SIGGRAPH Course*, 2006.
- [63] K. Grochow, S. Martin, A. Hertzmann, and Z. Popovic. Style-based inverse kinematics. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 23(3) :522–531, August 2004.
- [64] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [65] B. Guerville and M. Frenette. Mise au point sur la motion capture. *Sonovision magazine (Mocaplab)*, 534 :52–57, December 2008.
- [66] L. Han and L. Rudolph. Inverse kinematics for a serial chain with joints under distance constraints. In *Proc. of Robotics, Science, and Systems*, 2006.
- [67] S. Hauberg and K. Steenstrup-Pedersen. Predicting articulated human motion from spatial processes. *International Journal of Computer Vision*, 94 :317–334, 2011.
- [68] Timothy F Havel. Distance geometry. *eMagRes*, 1995.
- [69] C. Hecker, B. Raabe, R. W Enslow, J. DeWeese, J. Maynard, and K. van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Trans. on Graphics (TOG)*, 27(3) :1–11, 2008.
- [70] Bruce Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21 :65–84, 1992.
- [71] Bruce A Hendrickson. The molecule problem : Determining conformation from pairwise distances. Technical report, Cornell University, 1990.
- [72] J. Henry, H. Shum, and T. Komura. Environment-aware real-time crowd control. In *Symposium on Computer Animation SCA*, pages 193–200, July 2012.
- [73] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. *Computer Animation*, 0 :77, 2000.
- [74] Franck Hétroy. A discrete 3D+t Laplacian framework for mesh animation processing. Research Report RR-8003, INRIA, June 2012.
- [75] E. S.L Ho and T. Komura. Character motion synthesis by topology coordinates. In *Computer Graphics Forum*, volume 28, pages 299–308, 2009.
- [76] E. S.L Ho, T. Komura, and C. L Tai. Spatial relationship preserving character motion adaptation. *ACM Transactions on Graphics (TOG)*, 29(4) :1–8, 2010.

- [77] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.*, 29(4) :33 :1–33 :8, July 2010.
- [78] Edmond S. L. Ho, Taku Komura, and Chiew-Lan Tai. Spatial relationship preserving character motion adaptation. *ACM Trans. Graph.*, 29(4) :33 :1–33 :8, July 2010.
- [79] Jingtong Hou, Gregory E. Sims, Chao Zhang, and Sung-Hou Kim. A global representation of the protein fold space. *Proceedings of the National Academy of Sciences*, 100(5) :2386–2390, 2003.
- [80] Eugene Hsu, Sommer Gentry, and Jovan Popović. Example-based control of human motion. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pages 69–77, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.
- [81] Jianwei Hu, Ligang Liu, and Guozhao Wang. Dual laplacian morphing for triangular meshes. *Computer Animation and Virtual Worlds (Proceedings of CASA)*, 18(4-5) :271–277, 2007.
- [82] Jin Huang and Lu Chen. Efficient Mesh Deformation Using Tetrahedron Control Mesh. 1(212) :241–248, 2008.
- [83] Jin Huang, Xiaohan Shi, Xinguo Liu, Kun Zhou, Li-Yi Wei, Shang-Hua Teng, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Subspace gradient domain mesh deformation. *ACM SIGGRAPH 2006 Papers on - SIGGRAPH '06*, page 1126, 2006.
- [84] L. Ikemoto, O. Arikan, and D. Forsyth. Generalizing motion edits with gaussian processes. *ACM Trans. Graph.*, 28(1) :1–12, 2009.
- [85] Alec Jacobson, Ilya Baran, Jovan Popović, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 30(4) :78 :1–78 :8, July 2011.
- [86] A. Jaume-i Capó, J. Varona, M. González-Hidalgo, and F. Perales. Adding image constraints to inverse kinematics for human motion capture. *EURASIP J. Adv. Signal Process*, 2010 :4 :1–4 :13, January 2010.
- [87] Christophe Jermann, Dominique Michelucci, and Pascal Schreck. Modélisation géométrique par contraintes. In D. Bechmann and B. Péroche, editors, *Informatique graphique, modélisation géométrique et animation*, pages 185–210. Hermès Science, 2007.
- [88] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 26(3), July 2007.
- [89] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.*, 24(3) :561–566, July 2005.
- [90] Marko Jug, Janez Pers, Branko Dezman, and Stanislav Kovacic. Trajectory based assessment of coordinated human activity. In *3rd International Conference on Computer Vision Systems*, pages 534–543, 2003.

- [91] Roger E. Kahn, Michael J. Swain, Peter N. Prokopowicz, and R. James Firby. Gesture recognition using the perseus architecture. In *Conference on Computer Vision and Pattern Recognition*, pages 734–, 1996.
- [92] Zachy Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 279–286, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [93] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O’Sullivan. Geometric skinning with approximate dual quaternion blending. volume 27, page 105, New York, NY, USA, 2008. ACM Press.
- [94] Ryan Kennedy and Written Preliminary Examination II. Low-rank matrix completion. 2013.
- [95] Manmyung Kim, Kyunglyul Hyun, Jongmin Kim, and Jehee Lee. Synchronized multi-character motion editing. *ACM Trans. Graph.*, 28(3) :79 :1–79 :9, July 2009.
- [96] Oscar Kin-Chung Au, Chiew-Lan Tai, Ligang Liu, and Hongbo Fu. Dual laplacian editing for meshes. *IEEE Transactions on Visualization and Computer Graphics*, 12(3) :386–395, May 2006.
- [97] A.G. Kirk, J.F. O’Brien, and D.A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 782–788 vol. 2, 2005.
- [98] Hansjörg Klock and Joachim M. Buhmann. Multidimensional scaling by deterministic annealing. In *Proceedings EMMCVPR'97, Lecture Notes In Computer Science*, pages 245–260. Springer Verlag, 1997.
- [99] Lucas Kovar and Michael Gleicher. Automated extraction and parameterization of motions in large data sets. In *SIGGRAPH '04 : ACM SIGGRAPH 2004 Papers*, pages 559–568, NY, USA, 2004.
- [100] Vladislav Krayevoy and Alla Sheffer. Boneless motion reconstruction. (2) :2004, 2004.
- [101] R. Kulpa, F. Multon, and B. Arnaldi. Morphology-independent representation of motions for interactive human-like animation. *Computer Graphics Forum, Eurographics 2005 special issue*, 24(3) :343–352, 2005.
- [102] T. Kwon, K. H. Lee, J. Lee, and S. Takahashi. Group motion editing. *ACM Trans. on Graphics*, 27 :80 :1–80 :8, August 2008.
- [103] Carlile Lavor, Leo Liberti, and Nelson Maculan. An overview of distinct approaches for the molecular distance geometry problem. *Encyclopedia of Optimization, 2nd edn, Springer (to appear)*, 2008.
- [104] Jintae Lee. Notational representation of sign language : A structural description of hand configuration. In Wolfgang L. Zagler, Geoff Bushy, and Roland Wagner, edi-

- tors, *ICCHP*, volume 860 of *Lecture Notes in Computer Science*, pages 38–45. Springer, 1994.
- [105] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation : a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*, pages 165–172, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.
- [106] Lei Li, James McCann, Nancy Pollard, and Christos Faloutsos. Bolero : a principled technique for including bone length constraints in motion capture occlusion filling. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '10*, pages 179–188, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.
- [107] Xiubo Liang, Ludovic Hoyet, Weidong Geng, and Franck Multon. Responsive action generation by physically-based motion retrieval and adaptation. In Ronan Boulic, Yiorgos Chrysanthou, and Taku Komura, editors, *Motion in Games*, volume 6459 of *LNCS*, pages 313–324. Springer Berlin / Heidelberg, 2010.
- [108] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. *ACM Trans. Graph.*, 27(3) :78 :1–78 :10, August 2008.
- [109] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rossi, and Hans-Peter Seidel. Differential coordinates for interactive mesh editing. In *Shape Modeling Applications, 2004. Proceedings*, pages 181–190. IEEE, 2004.
- [110] Yaron Lipman, Olga Sorkine, Daniel Cohen-or, David Levin, Christian Rössl, and Hans peter Seidel. Differential coordinates for interactive mesh editing. In *In Proceedings of Shape Modeling International*, pages 181–190. Society Press, 2004.
- [111] Guodong Liu and Leonard McMillan. Estimation of missing markers in human motion capture. *The Visual Computer*, 22(9-11) :721–728, 2006.
- [112] Raymond Lockton and Andrew W. Fitzgibbon. Real-time gesture recognition using deterministic boosting. In *Proceedings of the British Machine Vision Conference 2002*, 2002.
- [113] Difei Lu, Ying Zhang, and Xiuzi Ye. A New Method of Interactive Marker-Driven Free form Mesh Deformation. *Geometric Modeling and Imaging–New Trends (GMAI'06)*, pages 127–134, 2006.
- [114] A.A. Maciejewski. Dealing with the ill-conditioned equations of motion for articulated figures. *IEEE Computer Graphics and Applications*, 10(3) :63–71, 1990.
- [115] Josiah Manson and Scott Schaefer. Hierarchical deformation of locally rigid meshes. *Computer Graphics Forum*, 30(8) :2387–2396, 2011.
- [116] Sebastian Martin, Christoph Huber, Peter Kaufmann, and Markus H Gross. Shape-preserving animation of deformable objects. In *VMV*, pages 65–72. Citeseer, 2009.

- [117] Hiroshi Masuda, Yasuhiro Yoshioka, and Yoshiyuki Furukawa. Preserving form features in interactive mesh deformation. *Computer-Aided Design*, 39(5) :361 – 368, 2007. <ce :title>Geometric Modeling and Processing 2006</ce :title> <xocs :full-name>Geometric Modeling and Processing 2006</xocs :full-name>.
- [118] Diana Mateus, Radu Horaud, David Knossow, Fabio Cuzzolin, and Edmond Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [119] Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. Efficient elasticity for character skinning with contact and collisions. *ACM Trans. Graph.*, 30(4) :37 :1–37 :12, July 2011.
- [120] Karl Menger. Untersuchungen über allgemeine metrik. *Mathematische Annalen*, 103(1) :466–501, 1930.
- [121] Mark Meyer, Alan Barr, Haeyoung Lee, and Mathieu Desbrun. Generalized barycentric coordinates on irregular polygons. *J. Graph. Tools*, 7(1) :13–22, November 2002.
- [122] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds, 2002.
- [123] Dominique Michelucci and H Lamure. Résolution de contraintes géométriques par homotopie. *Actes de AFIG 1994*, 1994.
- [124] Gary L. Miller, Dafna Talmor, Shang-Hua Teng, Noel Walkington, and Han Wang. Control volume meshes using sphere packing : Generation, refinement and coarsening. In *In Fifth International Meshing Roundtable*, pages 47–61, 1996.
- [125] J-S. Monzani, P. Baerlocher, R. Boulic, and D. Thalmann. Using an intermediate skeleton and inverse kinematics for motion retargetting. *Computer Graphics Forum*, 19(3), 2000. ISSN 1067-7055.
- [126] JorgeJ Moré and Zhijun Wu. Distance geometry optimization for protein structures. *Journal of Global Optimization*, 15(3) :219–234, 1999.
- [127] Antonio Mucherino, Carlile Lavor, Leo Liberti, and Nelson Maculan. Recent Advances on the Discretizable Molecular Distance Geometry Problem. *European Journal of Operational Research*, 219 :698–706, 2012.
- [128] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. Technical Report CG-2007-2, Universität Bonn, June 2007.
- [129] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. In *ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA)*, pages 137–146, Aire-la-Ville, Switzerland, 2006.



- [130] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. *ACM Trans. Graph.*, 24(3) :677–685, 2005.
- [131] Y. Nakamura and H. Hanafusa. Inverse kinematics solutions with singularity robustness for robot manipulator control. *Journal of Dynamic Systems, Measures and Control*, 108 :163–171, September 1986.
- [132] Andrew Nealen, Takeo Igarashi, Olga Sorkine, and Marc Alexa. Laplacian mesh optimization. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, GRAPHITE '06*, pages 381–389, New York, NY, USA, 2006. ACM.
- [133] Dan Overholt, John Thompson, Lance Putnam, Bo Bell, Jim Kleban, Bob Sturm, and JoAnn Kuchera-Morin. A multimodal system for gesture recognition in interactive music performance. *Computer Music Journal*, 33(4) :69–82, 2009.
- [134] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5) :1555–1564, 2010.
- [135] Maks Ovsjanikov, Jian Sun, and Leonidas Guibas. Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library, 2008.
- [136] Sang Il Park and Jessica K Hodgins. Capturing and animating skin deformation in human motion. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 881–889. ACM, 2006.
- [137] BRAIDO Peter and XUDONG ZHANG. Quantitative analysis of finger motion coordination in hand manipulative and gestic acts. *Human movement science*, 22(6) :661–678, 2004.
- [138] C.B. Phillips, J. Zhao, and N.I. Badler. Interactive real-time articulated figure manipulation using multiple kinematic constraints. In *Proc. of the 1990 symposium on Interactive 3D graphics (I3D'90)*, pages 245–250. ACM, New York, USA, 1990.
- [139] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.
- [140] D. Raunhardt and R. Boulic. Motion constraint. *The Visual Computer*, 25(5) :509 – 518, May 2009.
- [141] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace-beltrami spectra as "shape-dna" of surfaces and solids. *Computer-Aided Design*, 38(4) :342–366, 2006.
- [142] Damien Rohmer, Stefanie Hahmann, and Marie-Paule Cani. Exact volume preserving skinning with shape control. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation (SCA)*, pages 83–92, New Orleans, USA, 2009.

- [143] B. Roth. Rigid and flexible frameworks. *The American Mathematical Monthly*, 88(1) :pp. 6–21, 1981.
- [144] Raif M. Rustomov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, SGP '07, pages 225–233, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [145] J. B. Saxe. Embeddability of weighted graphs in k-space is strongly np-hard. 1979.
- [146] B. Schölkopf and A.J. Smola. *Learning with kernels : Support vector machines, regularization, optimization, and beyond*. the MIT Press, 2002.
- [147] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(4), 2005.
- [148] Jonathan Richard Shewchuk. Tetrahedral mesh generation by delaunay refinement. In *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG '98, pages 86–95, New York, NY, USA, 1998. ACM.
- [149] Xiaohan Shi. Mesh Puppetry : Cascading Optimization of Mesh Deformation with Inverse Kinematics. 26(3), 2007.
- [150] Robert H Shumway and David S Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4) :253–264, 1982.
- [151] Hang Si. *TetGen User Manual*, January 2006.
- [152] Manfred J Sippl and Harold A Scheraga. Solution of the embedding problem and decomposition of symmetric matrices. *Proceedings of the National Academy of Sciences*, 82(8) :2197–2201, 1985.
- [153] Atilla Sit. *Solving distance geometry problems for protein structure determination*. PhD thesis, Iowa State University, 2010.
- [154] Primoz Skraba, Maks Ovsjanikov, Frederic Chazal, and L Guibas. Persistence-based segmentation of deformable shapes. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 45–52. IEEE, 2010.
- [155] Wenhao Song and Ligang Liu. Stretch-based tetrahedral mesh manipulation. In *Proceedings of Graphics Interface 2007*, GI '07, pages 319–325, New York, NY, USA, 2007. ACM.
- [156] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '04, pages 175–184. ACM, 2004.
- [157] Olga Sorkine. *Laplacian mesh processing*. PhD thesis, Tel Aviv University, 2006.

- [158] Olga Sorkine. Least-squares rigid motion using svd. *Technical notes*, 120, 2009.
- [159] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, SGP '07, pages 109–116, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [160] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007.
- [161] D. A. Spielmat. Spectral partitioning works : planar graphs and finite element meshes. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, FOCS '96, pages 96–, Washington, DC, USA, 1996. IEEE Computer Society.
- [162] Nathan Srebro, Tommi Jaakkola, et al. Weighted low-rank approximations. In *ICML*, volume 3, pages 720–727, 2003.
- [163] Carsten Stoll, Edilson de Aguiar, Christian Theobalt, and Hans-Peter Seidel. A volumetric approach to interactive shape editing. Research Report MPI-I-2007-4-004, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, June 2007.
- [164] Robert W Sumner, Matthias Zwicker, Craig Gotsman, and Jovan Popović. Mesh-based inverse kinematics. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 488–495. ACM, 2005.
- [165] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.
- [166] S. Tak and H.-S. Ko. A physically-based motion retargeting filter. *ACM Trans. On Graphics (TOG)*, 24(1) :98–117, 2005.
- [167] S. Takahashi, K. Yoshida, T. Kwon, K. H. Lee, J. Lee, and S. Y. Shin. Spectral-based group formation control. *Computer Graphics Forum (Proc. Eurographics 2009)*, 28(2) :639–648, 2009.
- [168] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 351–358, New York, NY, USA, 1995. ACM.
- [169] Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pages 1345–1352, 2006.
- [170] Margara Tejera and Adrian Hilton. Space-time editing of 3d video sequences. In *Proceedings of the 2011 Conference for Visual Media Production*, CVMP '11, pages 148–157, Washington, DC, USA, 2011. IEEE Computer Society.
- [171] Warren S. Torgerson. *Theory and methods of scaling*. R.E. Krieger Pub. Co.

- [172] Lorenzo Torresani, Peggy Hackney, and Christoph Bregler. Learning motion style synthesis from perceptual observations. In *Advances in Neural Information Processing Systems*, pages 1393–1400, 2006.
- [173] Maxime Tournier, Xiaomao Wu, Nicolas Courty, Elise Arnaud, and Lionel Revéret. Motion compression using principal geodesics analysis. *Comput. Graph. Forum*, 28(2) :355–364, 2009.
- [174] L. Unzueta, M. Peinado, R. Boulic, and al. Suescun. Full-body performance animation with sequential inverse kinematics. *Graphical Models*, 70(5) :87–104, 2008.
- [175] VICON. Vicon motion system, 198x.
- [176] J. Vollmer, R. Mencl, and H. Müller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, pages 131–138, 1999.
- [177] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2) :283–298, 2008.
- [178] Kilian Q Weinberger, Fei Sha, Qihui Zhu, and Lawrence K Saul. Graph laplacian regularization for large-scale semidefinite programming. In *Advances in neural information processing systems*, pages 1489–1496, 2006.
- [179] C. Welman. Inverse kinematics and geometric constraints for articulated figure manipulation. Master’s thesis, Simon Frasier University, September 1993.
- [180] Yanlin Weng, Weiwei Xu, Yanchen Wu, Kun Zhou, and Baining Guo. 2d shape deformation using nonlinear least squares optimization. *Vis. Comput.*, 22(9) :653–660, September 2006.
- [181] Jane Wilhelms and Allen Van Gelder. Fast and easy reach-cone joint limits. *J. Graph. Tools*, 6(2) :27–41, September 2002.
- [182] W. A. Wolovich and H. Elliot. A computational technique for inverse kinematics. In *Proc. of 23rd IEEE Conf. on Decision and Control*, pages 1359–1363, 1984.
- [183] Di Wu and Zhijun Wu. An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data. *Journal of Global Optimization*, 37(4) :661–673, 2007.
- [184] Chuhua Xian, Hongwei Lin, and Shuming Gao. Automatic cage generation by improved obbs for mesh deformation. *Vis. Comput.*, 28(1) :21–33, January 2012.
- [185] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. Poisson shape interpolation. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, SPM ’05, pages 267–274, New York, NY, USA, 2005. ACM.
- [186] Masanobu Yamamoto, Takuya Kondo, Takashi Yamagiwa, and Kouji Yamanaka. Skill recognition. In *3rd International Conference on Face & Gesture Recognition*, pages 604–609, 1998.

- [187] K. Yamane and Y. Nakamura. Natural motion animation through constraining and de-constraining at will. *IEEE Tra. on Visualization and Computer Graphics*, 09(3) :352–360, 2003.
- [188] Xiaosong Yang, Arun Somasekharan, and Jian J Zhang. Curve skeleton skinning for human and creature characters. *Computer Animation and Virtual Worlds*, 17(3-4) :281–292, 2006.
- [189] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. *ACM SIGGRAPH 2004 Papers on - SIGGRAPH '04*, page 644, 2004.
- [190] S. Zhang, A. Nealen, and D. Metaxas. Skeleton based as-rigid-as-possible volume modeling. *Eurographics Association, Norrköping, Sweden*, pages 21–24, 2010.
- [191] J. Zhao and N. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Tra. on Graphics (Proc. SIGGRAPH)*, 13(4) :313–336, 1994.
- [192] Yong Zhao. Rigidity Constraints for Large Mesh Deformation. 24(60603078) :47–55, 2009.
- [193] Yong Zhao and Jianhui Liu. Volumetric subspace mesh deformation with. (September) :519–532, 2012.
- [194] Kun Zhou, Jin Huang, John Snyder, Xinguo Liu, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Large mesh deformation using the volumetric graph laplacian. *ACM Trans. Graph.*, 24(3) :496–503, July 2005.
- [195] Xiang Sean Zhou and Thomas S. Huang. Comparing Discriminating Transformations and SVM for Learning During Multimedia Retrieval. In *ACM Multimedia*, pages 137–146, 2001.
- [196] Michael Zollhöfer, Ezgi Sert, Günther Greiner, and Jochen Süßmuth. Gpu based arap deformation using volumetric lattices. In Carlos Andujar and Enrico Puppo, editors, *Eurographics (Short Papers)*, pages 85–88. Eurographics Association, 2012.
- [197] Victor Brian Zordan and Nicholas C. Van Der Horst. Mapping optical motion capture data to skeletal motion using a physical model. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '03*, pages 245–250, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.







## Méthode de résolution de Newton-Raphson

Soit le problème  $F$  d'optimisation global constitué de d'un ensemble de contraintes  $f \in E$  non linéaire :

$$\min_{V=(v_1, \dots, v_n)} \sum_{f \in E} f(V). \quad (\text{A.1})$$

Une façon de trouver la solution de ce problème est d'utiliser une descente de gradient. Plusieurs possibilités existent, comme la méthode connue de Newton-Raphson. La méthode de Newton-Raphson résout itérativement l'équation  $f(V) = 0$ , avec  $V$  un vecteur de  $\mathbb{R}^n$  et  $f$  une fonction continue et dérivable. Un exposé approfondi est donné par Bjorck [18].

**La méthode de Newton-Raphson appliqué à une variable.** Soit une variable  $x \in \mathbb{R}$ . En partant d'une approximation  $x_0$  de la solution, le principe est de calculée itérativement la séquence de valeurs pour  $x : x_1, x_2, x_3 \dots$  afin d'approcher la solution.  $x_{n+1}$  est calculée de la façon suivante : nous calculons la tangente à la fonction  $f$  au point  $(x_n, f(x_n))$  et  $x_{n+1}$  est l'abscisse du point intersection de cette tangente et de l'axe des  $x$ .

En considérant une fonction  $f$  continûment dérivable sur un intervalle  $I$  de  $\mathbb{R}$  et  $x_0 \in I$ , mathématiquement on peut estimer la valeur d'un point sur  $f$  en se basant sur le développement de Taylor au premier ordre :

$$f(x) \cong f(x_0) + f'(x_0)(x - x_0). \quad (\text{A.2})$$

Nous cherchons donc à résoudre le problème

$$0 = f(x_0) + f'(x_0)(x - x_0), \quad (\text{A.3})$$

ce qui est équivalent à :

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}. \quad (\text{A.4})$$

Finalement on cherche à partir d'une première approximation  $x_0$ , à trouver une racine de  $f$  par



récurrance de la suite :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \quad (\text{A.5})$$

**Généralisation à l'ensemble du problème.** On peut généraliser la méthode à plusieurs variables. Soit  $F$  une application de  $\mathbb{R}^m$  dans  $\mathbb{R}^{3*p}$  ( $p \geq 1$ ) et  $\mathbf{V} \in \mathbb{R}^{3*p}$ , nous pouvons reformuler le problème précédent par :

$$\mathbf{V}_{k+1} = \mathbf{V}_k - J(\mathbf{V}_k)^{-1}F(\mathbf{V}_k), \quad (\text{A.6})$$

où  $J(\mathbf{V}_k)$  est la matrice jacobienne de  $F$  appliquée à  $\mathbf{V}_k$  :

$$J_{ij}(\mathbf{V}_k) = \frac{\partial f_i}{\partial v_{j,l}}(\mathbf{V}_k), \quad 1 \leq i \leq m, \quad 1 \leq (j*l) \leq 3*p, \quad (1 \leq l \leq 3). \quad (\text{A.7})$$

Enfin  $F$  peut être mis sous la forme d'un système regroupant l'ensemble de  $m$  contraintes de distances  $F = (f_1, \dots, f_m)$  tel que

$$F : \begin{cases} f_1(\mathbf{v}_1, \dots, \mathbf{v}_n) = 0 \\ \vdots \\ f_p(\mathbf{v}_1, \dots, \mathbf{v}_n) = 0 \end{cases} \quad (\text{A.8})$$

Dans le cas où la matrice jacobienne  $J$  est non singulière, ce système peut être résolu par des méthodes numériques directes (Gauss, Gauss-Jordan, ...) ou itératives (Gauss-Seidel Jacobi). Toutefois dans notre cas,  $J$  est sans à priori singulière ( $n \neq p$ ), on utilise donc la matrice pseudo inverse  $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$  de Moore-Penrose calculable par une décomposition en valeur singulière (SVD).

Q : si il existe plusieurs solutions à  $F(x) = 0$ , converger vers la meilleure solution au problème c'est fournir un  $x_0$  plus proche de la solution voulue que les autres.

# B

## Technique *As-Rigid-As-Possible*

Sorkine [159] propose la technique *As-Rigid-As-Possible*(ARAP) pour éditer les maillages surfaciques impliquant de grandes déformations. Basée sur une méthode itérative, l'auteur montre que l'intérêt de la technique *As-Rigid-As-Possible* réside dans la préservation entre chaque itération de la rigidité propre à chaque cellule du maillage. La figure B.1 illustre bien un tel principe.

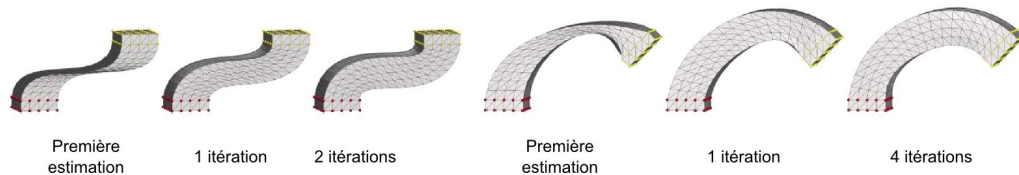


FIGURE B.1 – Exemple d'édition de maillage utilisant *ARAP* [159]. La première estimation est le résultat de la minimisation décrite par l'équation B.2.

### B.1 Principe

Soit  $\mathcal{M}$  le maillage triangulé constitué de  $n$  sommets et  $m$  triangles. Notons que  $\mathcal{N}(i)$  est l'ensemble des points connectés au sommet  $v_i$ . La géométrie de  $\mathcal{M}$  est décrite par un ensemble de positions  $\{v_i\} \in \mathbb{R}^3$ . La déformation de  $\mathcal{M}$  est donnée par  $\mathcal{M}'$  dont l'ensemble des sommets déformés sont  $\{v'_i\}$ .

**Rappels sur l'édition d'un graphe avec l'opérateur Laplacien.** Nous considérons l'ensemble des coordonnées différentielles  $\{\delta_i\}$ .  $\delta_i(\in \mathbb{R}^3)$  est le Laplacien de  $v_i$  et plus précisément la coordonnée d'indice  $i$  est représentée par la différence entre  $v_i$  et la moyenne de ses

voisins :

$$\delta_i = \mathcal{L}(\mathbf{v}_i) = \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{v}_i - \mathbf{v}_j), \quad (\text{B.1})$$

où  $\mathcal{L}$  L'opérateur Laplacien discret appliqué à  $\mathbf{v}_i$ .  $w_{ij}$  est le poids à associer aux arêtes de  $\mathcal{M}$ . En fixant par substitution dans le problème précédent un ensemble de sommets  $\{\mathbf{u}_i\}$  tel que  $i \in \{1 \dots n\}$  et  $l < n$ , la fonction erreur  $E_U$  qui permet de trouver la nouvelle configuration  $\mathcal{M}'$  peut être exprimée par :

$$E_U(\mathcal{M}') = \sum_{i=1}^n \|\delta_i - \mathcal{L}(\mathbf{v}'_i)\|^2 + \sum_{i=m}^n \|\mathbf{v}'_i - \mathbf{u}_i\|^2. \quad (\text{B.2})$$

Cette formulation du problème d'édition de maillage est limitée pour des transformations avec des rotations de grandes amplitudes. Cela s'explique car l'opérateur Laplacien ne préserve pas localement les rotations. Ce point est illustré par les premières estimations dans la figure B.1. Une manière de remédier à ce problème est d'introduire localement une rotation appliquée à chaque différentielle  $\{\delta_i\}$ .

**Analyse de la transformation rigide entre deux cellules** Soit la cellule  $C_i$  appartenant à  $\mathcal{M}$  et sa version transformé  $C'_i$  appartenant à  $\mathcal{M}'$ . En faisant l'hypothèse qu'il existe une transformation rigide entre  $C_i$  (dans sa configuration de référence) et  $C'_i$  (cellule déformée), l'idée est de déterminer itérativement la meilleure approximation de la rotation  $\mathbf{R}_i$  appliquée à  $C_i$  après déformation du maillage :

$$E_C(C_i, C'_i) = \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)\|^2. \quad (\text{B.3})$$

Dans cette équation  $w_{ij}$  est le poids cotangent associé aux sommets  $i$  et  $j$ .

**Énergie de déformation globale** Nous pouvons généraliser l'équation précédente pour le calcul de l'énergie de déformation globale du maillage entre deux déformations par la fonction quadratique :

$$E(\mathcal{M}') = \sum_{i=1}^n E(C_i, C'_i), \quad (\text{B.4})$$

ce qui est équivalent à :

$$E(\mathcal{M}') = \sum_{i=1}^n E(C_i, C'_i) = \sum_{i=1}^n \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)\|^2 \quad (\text{B.5})$$

**Algorithme** Le problème formulé précédemment est clairement non linéaire. L'auteur propose donc d'utiliser une descente de gradient pour minimiser  $E(\mathcal{M}')$ . A partir d'une première estimation des coordonnées  $\{\mathbf{v}'_i\}$ , la technique repose sur un algorithme par alternance :

1. calcul d'une première estimation des coordonnées  $\{\mathbf{v}'_i\}$  minimisant  $E_U(\mathcal{M}')$  (équation (B.2)) ;
2. calcul des transformations  $\mathbf{R}_i$  à partir de l'équation (B.3) ;
3. calcul des coordonnées  $\{\mathbf{v}'_i\}$  minimisant  $E(\mathcal{M}')$ .

L'algorithme boucle sur les phases 2 et 3. Nous détaillons dans les deux sections suivantes ces deux phases de l'algorithme.

## B.2 Calcul des rotations $R_i$

Pour trouver la rotation  $\mathbf{R}_i$ , l'auteur utilise une *décomposition polaire*, technique (décrite par [158]) connue et développée par Arun & al [8]. Cette technique consiste à effectuer une décomposition en valeurs propres (SVD) de la matrice de covariance  $\mathbf{S}_i$  de la cellule  $i$  telle que :

$$\mathbf{S}_i = \sum_{j \in \mathcal{N}(i)} w_{ij} \mathbf{e}_{ij} \mathbf{e}_{ij}'^T = \mathbf{P}_i \mathbf{D}_i \mathbf{P}_i. \quad (\text{B.6})$$

Dans cette équation  $\mathbf{e}_{ij} := \mathbf{v}_i - \mathbf{v}_j$  et  $\mathbf{e}_{ij}' := \mathbf{v}'_i - \mathbf{v}'_j$  désignent respectivement les vecteurs allant du sommet  $i$  à son sommet voisin  $j$  avant et après déformation.  $\mathbf{D}_i$  est la matrice diagonale contenant les valeurs des poids  $w_{ij}$ ,  $\mathbf{P}_i$  est la matrice de dimension  $3 \times |\mathcal{N}(i)|$  dont les colonnes contiennent les vecteurs  $\mathbf{e}_{ij}$ . Ainsi à partir de la décomposition en valeurs propres de  $\mathbf{S}_i$  telle que  $\mathbf{S}_i = \mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^T$ , on obtient que  $\mathbf{R}_i = \mathbf{V}_i \mathbf{U}_i^T$  qui est de dimension  $3 \times 3$ .

## B.3 Minimisation de l'énergie $E$

Pour calculer les positions  $\{\mathbf{v}'_i\}$  optimales à partir des rotations, nous déterminons l'équation pour laquelle le gradient de  $E(\mathcal{M}')$  en fonction des  $\{\mathbf{v}'_i\}$  s'annule.

$$\begin{aligned} \frac{\partial E(\mathcal{M}')}{\partial \mathbf{v}'_i} &= \frac{\partial}{\partial \mathbf{v}'_i} \left( \sum_{j \in \mathcal{N}(i)} w_{ij} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)\|^2 + \sum_{j \in \mathcal{N}(i)} w_{ji} \|(\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_j(\mathbf{v}_i - \mathbf{v}_j)\|^2 \right) \\ &= \sum_{j \in \mathcal{N}(i)} 2w_{ij} ((\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_i(\mathbf{v}_i - \mathbf{v}_j)) + \sum_{j \in \mathcal{N}(i)} 2w_{ji} ((\mathbf{v}'_i - \mathbf{v}'_j) - \mathbf{R}_j(\mathbf{v}_i - \mathbf{v}_j)). \end{aligned}$$

En utilisant  $w_{ij} = w_{ji}$ , nous obtenons

$$\frac{\partial E(\mathcal{M}')}{\partial \mathbf{v}'_i} = \sum_{j \in \mathcal{N}(i)} 4w_{ij}((\mathbf{v}'_i - \mathbf{v}'_j) - \frac{1}{2}(\mathbf{R}_i + \mathbf{R}_j)(\mathbf{v}_i - \mathbf{v}_j)). \quad (\text{B.7})$$

Puis en posant que la dérivée partielle est égale à zéro pour chaque  $\mathbf{v}'_i$ , nous obtenons le système d'équations linéaires éparse :

$$\sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{v}'_i - \mathbf{v}'_j) = \sum_{j \in \mathcal{N}(i)} \frac{w_{ij}}{2}(\mathbf{R}_i + \mathbf{R}_j)(\mathbf{v}_i - \mathbf{v}_j). \quad (\text{B.8})$$

Finalement en identifiant que le premier terme est équivalent à l'opérateur de Laplace Beltrami discret associé au sommet  $i$ , nous pouvons généraliser la démarche à l'ensemble du maillage par une notation plus simplifiée par la fonction quadratique  $E$  :

$$E(\mathcal{M}') = \sum_{i=1}^n \|\delta_i^A - \mathcal{L}(\mathbf{v}'_i)\|^2 \quad (\text{B.9})$$

où  $\delta_i^A$  est le terme correspondant au membre de gauche de l'équation (B.8) évalué à chaque itération.



## Résumé

L'animation de personnages virtuels guidée par des données fait l'objet de nombreuses études dans le domaine de l'informatique graphique. Dans ce contexte, le mouvement est classiquement défini par une suite de squelettes au cours du temps, chacun étant décrit par un vecteur de positions et de rotations. Le maillage 3D est ensuite guidé par les squelettes au moyen d'un couplage défini explicitement. L'enchaînement des différentes étapes de ce processus est difficile à mettre en œuvre, et conduit à des approximations et des erreurs de modélisation, à la fois au niveau de l'animation du squelette et de la déformation du maillage.

Dans ce manuscrit nous proposons d'étudier d'autres représentations du mouvement par le biais d'un ensemble de relations spatiales. Cette approche nous permet de tenir compte implicitement de contraintes de distance entre les points de la structure articulée, du maillage et de l'environnement, et d'exprimer en particulier la notion de contact. Deux axes d'étude sont principalement abordés : le premier considère le mouvement dans l'espace métrique, et le second caractérise chaque posture par son information différentielle dans l'espace Laplacien.

Dans un premier temps, nous proposons de représenter les squelettes associés aux postures du mouvement par un ensemble de distances. Caractériser une telle structure dans l'espace métrique se ramène à un formalisme mathématique connu sous le nom de problème de la géométrie des distances. Ainsi, nous nous inspirons des différentes techniques existantes et les appliquons au contrôle du mouvement. L'objectif est de produire de nouveaux mouvements à partir de processus d'édition ou d'inversion cinématique. Nous montrons que cette représentation permet un contrôle simple et intuitif de l'animation d'un personnage. Elle possède également plusieurs propriétés exploitables dans le cadre de l'analyse du mouvement. Ce dernier point est illustré par une application originale de recherche de mouvements dans des grandes bases de données.

Dans un second temps, nous définissons le mouvement par un ensemble de graphes dont les sommets sont caractérisés par une information différentielle. A travers cette représentation, nous proposons une nouvelle méthode d'édition du mouvement couplant des contraintes de distance avec l'opérateur Laplacien discret. Cet opérateur permet de préserver les relations spatiales lors de l'édition du mouvement alors que les contraintes de distance préservent certaines propriétés inhérentes au squelette. Ce concept donne lieu à plusieurs applications dédiées à la reconstruction et l'édition de mouvement : (i) l'édition interactive d'animation de squelette, où nous proposons d'éditer tout type de mouvement avec de fortes déformations tout en préservant l'information spatio-temporelle ; (ii) la reconstruction de trajectoires de marqueurs : en faisant l'hypothèse qu'il existe un lien entre la trajectoire d'un marqueur et celles de ses voisins, nous proposons de reconstruire les trajectoires incomplètes ; (iii) l'animation de maillage où nous proposons un nouveau processus d'animation directement guidé par les trajectoires des marqueurs.

**Mots clés:** Edition de mouvement, Génération de mouvement, Opérateur Laplacien, Mouvement dans l'espace métrique, Informatique graphique.

## Abstract

The animation of virtual characters driven by data is one of the key topics in computer graphics. In this context, a motion is classically defined by a list of skeletons over time, each of them is described by a vector of positions and rotations. The 3D mesh is then controlled by the skeletons by a rigging step between the skeleton and the mesh.

In this document, we propose to study other representations of the motion through a set of spatial relationships. Two approaches are proposed : the first considers the motion in the metric space and the second characterizes each posture by a differential representation using the Laplacian operator.

First, we propose to represent the postures of the motion by a set of distances. The goal is to produce new motions from an editing process or kinematic inversions. We show that this representation allows a simple and intuitive control of the animation of a character. It also has several properties exploitable in the context of motion analysis. This last point is illustrated by an original application of motion retrieval in large databases.

Next, we define the motion by a set of graphs. The vertices are characterized by a differential information. Through this representation, we propose a new method to edit a motion coupling constraints of distance and the use the discrete Laplacian operator. This operator preserves the spatial relationships during the edition of the motion and the constraints of distance preserve the properties of the induced skeleton. This concept allows us to propose three applications dedicated to the reconstruction and the edition of motions : (i) an interactive system to edit sequences of skeletons using the discret 3D+ $t$  Laplacian operator ; (ii) reconstruction of marker trajectories (iii) animation of mesh constrained by an implicit skeleton and driven by markers trajectories.



n d'ordre : 000000000

**Université de Bretagne Sud**

Centre d'Enseignement et de Recherche Y. Coppens - rue Yves Mainguy - 56000 VANNES

Tèl : + 33(0)2 97 01 70 70 Fax : + 33(0)2 97 01 70 70