HAL
open science

Animation de personnages par skinning à volume constant

Damien Rohmer

To cite this version:

Damien Rohmer. Animation de personnages par skinning à volume constant. Synthèse d'image et réalité virtuelle [cs.GR]. 2007. hal-01017921

HAL Id: hal-01017921

https://inria.hal.science/hal-01017921

Submitted on 3 Jul 2014

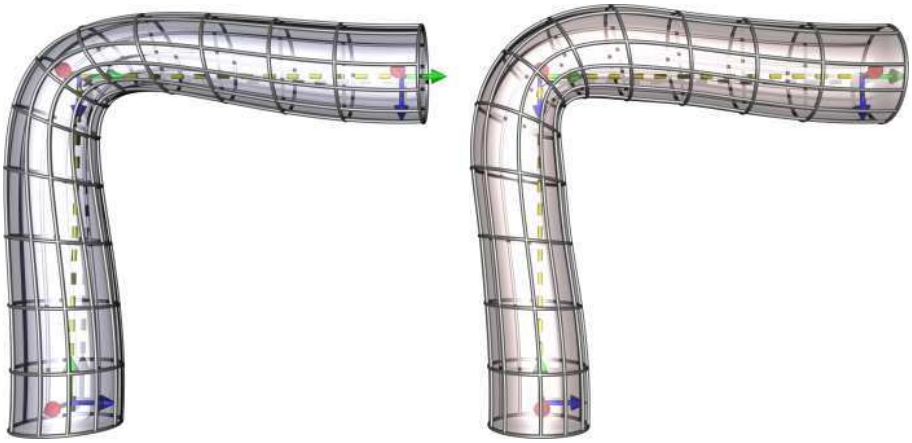
HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

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.

	Rapport de Stage	*Dates*
	Stage de Projet de fin d'études	du 1er mars 2007 au 31 août 2007
		Année 2007

Rohmer Damien
Spécialité ETI, Image
Non Confidentiel

Animation de Personnages par skinning à volume constant



INRIA–EVASION, Montbonnot, France
Laboratoire Jean-Kuntzmann, Grenoble, France

Encadrants:
Hahmann Stefanie (LJK, Grenoble)
Cani Marie-Paule (INRIA–EVASION, Montbonnot)
Enseignants-chercheurs INPG

Table des matières

1 Introduction **3**

2 Animations de corps de personnages **5**
2.1 Généralités . . . 5
2.1.1 Origine de la déformation . . . 5
2.1.2 Méthodes Physiques . . . 6
2.1.3 Cas des méthodes non physiques classiques . . . 7
2.2 Méthode du Skinning . . . 10
2.2.1 Déformation d'un squelette . . . 10
2.2.2 Assemblage de blocs . . . 11
2.2.3 Skinning simple . . . 13
2.2.4 Soft skinning . . . 13
2.2.5 Difficultés, Choix des poids . . . 15
2.3 Solutions existantes pour le skinning . . . 16
2.3.1 Modification des poids et limitation en angle . . . 16
2.3.2 Ajout de repères . . . 17
2.3.3 Méthode multi-poids . . . 17
2.3.4 EigenSkin . . . 17
2.3.5 Axe médian . . . 18
2.3.6 Construction d'un squelette flexible . . . 18
2.3.7 Approche choisie . . . 18

3 Contrainte de volume **19**
3.1 Expression d'un domaine bordé par une surface fermée . . . 19
3.1.1 Cas général . . . 19
3.1.2 Cas triangulé . . . 20
3.1.3 Interprétation . . . 21
3.1.4 Utilisation de la trilinéarité . . . 22
3.2 Méthodes existantes de conservation de volume . . . 23
3.2.1 Surfaces implicites . . . 23
3.2.2 Free Form deformation . . . 24
3.2.3 Cas des surfaces multirésolution . . . 25

4 Application de la contrainte de volume au skinning **30**
4.1 Exemple du choix des poids de skinning . . . 30
4.1.1 Modèle du cylindre . . . 30
4.1.2 Équation de la surface skinnée . . . 30
4.1.3 Calcul du volume . . . 34
4.1.4 Poursuite des travaux . . . 35
4.2 Méthode générale de correction de volume . . . 35
4.3 Cas linéarisé . . . 36

4.4 Cas exact . . . 37
4.4.1 Résultats . . . 37
4.4.2 Discussion . . . 40
4.5 Pondération locale de la déformation . . . 41
4.5.1 Résultats . . . 43
4.5.2 Discussion . . . 44
4.6 Contrainte de position fixe . . . 45
4.7 Direction Normale . . . 46
4.8 Ajout des coordonnées laplaciennes . . . 49

5 Travaux Futurs **51**

6 Conclusion **54**

A Correction de volume sur les trois axes dans le cas linéarisé **55**

B Correction de volume sur les trois axes dans le cas exact **57**
B.1 Séparation des étapes . . . 57
B.2 Résolution . . . 58

C Correction de volume pondéré sur les sommets dans le cas exact **60**

D Contrainte de positions fixes **62**
D.1 Minimisation à résoudre . . . 62
D.2 Résolution . . . 63

E Correction de volume dans la direction normale dans le cas linéarisé **65**

F Correction de volume dans le cas des coordonnées laplaciennes **67**

Résumé

La déformation de personnages animés par la méthode de skinning est largement répandue dans le domaine du cinéma et du jeu vidéo grâce à son contrôle intuitif et à son évaluation rapide. Cependant, pour certains mouvements, l'aspect réaliste de l'animation peut être mis en défaut par la perte d'une partie du volume du personnage déformé. Ce rapport présente une méthode de contrainte de volume appliquée à la déformation d'un objet à l'aide d'un squelette d'animation.

Nous procédons en une première étape de déformation par skinning classique, puis corrigeons le volume de la surface obtenue dans un second temps. La correction est obtenue par une minimisation sous contrainte dont les paramètres sont liés aux seules données du skinning.

Une solution analytique est obtenue, permettant une exécution de l'algorithme à plus de 30 images par seconde. Le volume du personnage ainsi déformé peut alors être corrigé de façon exacte dans le cas d'un traitement séparé des axes. Une méthode approchée est également mise en place pour permettre une correction liée localement à la forme de l'objet animé.

La méthode utilisée permet une déformation à volume constant tout en préservant la simplicité et la rapidité d'exécution du skinning. Celle-ci est intrinsèquement liée aux paramètres de la déformation par squelette et ne recquiert pas de travail supplémentaire de l'artiste. Enfin, différents effets visuels peuvent être obtenus selon l'optimisation utilisée.

Abstract

Animated characters obtained with the skinning method are widely used in movies and video games industry due to its intuitive control and its fast implementation. However, when this method is used for large deformations, the realistic beavior of the animation might be spoiled by the loss of a certain amount of volume of the animated object. Therefore, a method that constraints the volume of a character deformed with an underlying skeleton is set-up.

The correction is performed in two steps. First, the surface is deformed with a classical skinning algorithm. Next, the surface is corrected in order to keep its volume constant. The correction is given by a constrained minimisation where the parameters only depend on the skinning data. The method of Lagrange multipliers is used is order to solve the equations.

An analytical solution is given that enables the execution at interactive speed. An exact correction of the volume can be applied on the character when the axes are treated separatedly. A correction that locally depend on the shape of the object is also carried on via an approximated correction. The error of the approximated volume is less than 5% in the worst case.

This method enables a deformation with a constant volume while keeping the advantage of the skinning. Moreover, the correction is linked to the parameters that are already given for a deformation performed with an animation-skeleton. Therefore, there is no need of any extra work for the artist. Finally, different visual effects can be obtained depending on the type of function that is optimized.

Remerciements

Je souhaite tout d'abord remercier mes encadrants de stage Marie-Paule CANI et Stefanie HAHMANN de m'avoir accordé leur confiance afin de me permettre de travailler sur ce sujet de stage malgré mon cursus un peu différent des cas habituels.
Je remercie également l'équipe du LJK pour m'accueillir dans leur locaux ainsi que celle d'EVASION de l'INRIA. En particulier Paul KRY pour son intérêt dans mon travail.
De même, je remercie Christine DEPRAZ pour m'avoir fourni les modèles artistiques animés pouvant nous servir lors de nos tests.
Enfin, je remercie mes collègues de bureaux pour l'ambiance chaleureuse partagée jusqu'à présent.

Notations utilisées

Nous précisons quelques notations que l'on utilisera dans ce document. Les conventions choisies dans cette partie ne seront pas forcément précisées à chaque fois dans le reste du texte.

Nous utilisons les lettres en gras pour désigner un vecteur. Et la $i^{\text{ème}}$ composante d'un vecteur est alors désignée par la lettre simple indicée par i. Ainsi la $i^{\text{ème}}$ composante du vecteur $\mathbf{a}$ sera désignée par a_i. Il conviendra de ne pas confondre avec le $i^{\text{ème}}$ vecteur d'une famille désigné dans ce cas par $\mathbf{a}_i$ par exemple.

Ce travail se place dans l'espace $\mathbb{R}^3$. Pour simplifier les notations, nous passerons donc souvent de l'expression vectorielle à l'expression scalaire sur chaque axe. Nous conviendrons qu'un vecteur $\mathbf{b} \in \mathbb{R}^3$ pourra également se désigner par ses trois composantes $\mathbf{b} = (b^x, b^y, b^z)$.

Nous aurons cependant recours également à des vecteurs de dimensions $3\,N_s$ où N_s désignera le nombre de sommets utilisés pour une surface. Ces vecteurs sont ordonnés suivant l'indice des sommets et suivant les axes (x, y, z). Considérons un vecteur $\mathbf{c} \in \mathbb{R}^{3\,N_s}$. Nous désignons alors par $(\mathbf{c}^x, \mathbf{c}^y, \mathbf{c}^z)$ les vecteurs de $\mathbb{R}^{N_s}$ correspondant à chacun des axes. Par contre, et lorsque cela ne prêtera pas à confusion, le $i^{\text{ème}}$ vecteur de $\mathbb{R}^3$ formé par (c_i^x, c_i^y, c_i^z) sera simplement noté $\mathbf{c}_i$.

D'une façon générale, la position spatiale des sommets sera désignée par le vecteur $\mathbf{x}$. Par souci de confort lors de la lecture, ces composantes seront désignées spécialement par $\mathbf{x} = (x, y, z)$.

Les matrices seront désignées par des lettres majuscules droites comme par exemple M dont les composantes sont notées par M_{ij}.

Le produit scalaire entre deux vecteurs $\mathbf{a}$ et $\mathbf{b}$ sera indifféremment noté $< \mathbf{a}, \mathbf{b} >$ ou $\mathbf{a} \cdot \mathbf{b}$ suivant les cas, dans l'espoir de faciliter la lisibilité de l'équation.

Enfin, on considère des surfaces formées par N_s sommets. À plusieurs reprises, les formules feront appel à des sommations sur l'ensemble des sommets de ces surfaces. Dans ce cas non litigieux, les bornes de la sommation ne seront pas rappelées afin de ne pas surcharger les notations. Ainsi l'expression $\sum_i a_i = \sum_{i=1}^{N_s} a_i$. Et de même, pour des sommations sur plusieurs indices, on notera $\sum_{(i,j)} a_{ij} = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} a_{ij}$. De même, lorsque nous ferons référence à un ensemble de composantes sans préciser les bornes de variation des indices, il sera entendu que ces indices varieront sur l'ensemble des sommets. Ainsi l'expression $\forall i$ sans plus de précisions désignera $\forall i \in [\![1, N_s]\!]$ lorsqu'aucune ambiguité ne sera possible.

Chapitre 1

Introduction

Aujourd'hui, l'animation de personnages (humains ou animaux) est devenue courante dans le domaine de l'imagerie de synthèse. Cependant, une étude attentive des déformations mises en place pour le corps du personnage montre que celles-ci sont parfois encore très simples ce qui risque quelquefois de nuire à leur réalisme. Le cas le plus évident étant celui des déformations calculées en temps réel comme dans les jeux vidéo où l'on déduit directement le mouvement de la peau à partir d'un squelette d'animation. Le problème n'est cependant pas facile à régler car ces mouvements sont induits par celui des muscles. Or les équations constitutives régissant les déformations de ces muscles sont complexes. Le cas d'application des déformations dans une optique de temps réel nécessite donc des simplifications importantes. Cependant il est intéressant de rester cohérent par rapport à la physique sous-jacente afin de garder un comportement réaliste et robuste pour les cas les plus larges possible.

Dans notre cas, nous allons plus particulièrement nous intéresser aux cas de personnages animés à l'aide d'un squelette interne. Cette méthode d'animation possédant l'avantage d'être intuitive et pouvant être implémentée en temps réel possède cependant quelques désavantages visuels nuisant à l'aspect réaliste du résultat lorsque les déformations sont importantes. Ces problèmes étant souvent liés à la perte de volume d'une partie du personnage déformé, l'objectif de ce travail est d'assurer une déformation du corps à volume constant, dans le but d'améliorer le réalisme visuel. Son aspect novateur proviendra de l'application locale de la contrainte d'un volume constant spécifique à la déformation obtenue par skinning et à ces paramètres.

Nous tenterons également de toujours garder la possibilité d'implémenter la contrainte de volume dans un cadre temps réel.

Dans un premier temps, un rappel sera fait sur quelques méthodes classiques d'animation de personnages. Une explication plus détaillée aura lieu sur la méthode de skinning par squelette que l'on cherchera à améliorer. Après avoir pu constater ses avantages et inconvénients, nous donnerons quelques pistes déjà étudiées dans la littérature visant à améliorer cette méthode de déformation. Compte tenu de leur aspect plus ou moins général, nous tenterons de considérer les plus appropriés dans notre cas.

Les parties suivantes de l'étude s'intéresseront maintenant plus particulièrement au calcul du volume. Afin de mettre en place notre méthode de conservation de volume, nous établirons les équations utilisées pour le calcul du volume d'une surface triangulée. Dans la suite, nous ferons un point sur les différentes méthodes déjà utilisées dans la littérature afin de réaliser des déformations à volume constant.

Enfin, la partie suivante présentera les résultats concernant notre correction de volume appliquée au skinning. Nous présenterons tout d'abord une résolution linéarisée puis exacte de l'intégration de la contrainte de volume en utilisant l'avantage de la trilinéarité de l'expression du volume. Nous tenterons ensuite d'améliorer l'aspect visuel recherché en pondérant la déformation suivant les régions de la surface à déformer, ainsi qu'en ajoutant des contraintes de positions fixes.

Par la suite, la méthode de correction de volume sera améliorée en considérant un déplacement s'adaptant localement à la normale de la surface skinnée, bien que l'aspect trilinéraire soit perdu. L'expression théorique de la correction en prenant en compte une pondération suivant les coordonnées laplaciennes sera enfin mise en place de façon à prendre en compte l'aspect du voisinage lors de la correction. Nous présenterons finalement les futures pistes de recherche que nous pensons suivre pour continuer et étendre ce travail.

Chapitre 2

Animations de corps de personnages

2.1 Généralités

On peut généralement classer les méthodes d'animations d'objets déformables en deux catégories.

- Les méthodes basées sur la physique des déformations et donc prenant en compte la dynamique des forces qui interagissent avec l'objet
- Les méthodes basées simplement sur la cinématique des déformations qui permettent une animation à moindre coût mais souvent moins correcte de par le manque de prise en compte des lois de mouvement.

Bien que la méthode d'animation prise en compte dans ce travail ne soit pas basée sur la théorie physique des déformations du corps du personnage, il peut cependant être intéressant d'en connaître quelques aspects.

2.1.1 Origine de la déformation

L'aspect de la déformation qui nous intéresse pour l'animation de personnages est celle de la peau. Les déformations internes ne sont, quant à elles, pas visualisées et ne nécessitent donc pas d'être obligatoirement connues. Cependant, ce sont les muscles qui sont à l'origine du mouvement de par leurs contractions. Les déformations des muscles sont également la cause de celles de la peau qui glisse sur eux. Enfin les mouvements des muscles sont transmis par le squelette qui va contraindre ce déplacement à seulement certains degrés de liberté.

Cas de la peau

La peau est l'élément le plus important car c'est elle que l'on va visualiser, il est donc important de connaître les propriétés basiques de celle-ci. La peau peut être vue à l'échelle macroscopique comme une surface continue possédant un comportement visco-elastique. Elle consiste en une structure de collagène et d'élastine dont l'enchevêtrement lui donne ses propriétés mécaniques [WKT97, EMBL99, TCET05]. On peut alors considérer qu'au niveau macroscopique, un élément de peau possède un comportement incompressible et anisotrope [Hen01] d'où la création de plis dans des directions privilégiées lors des mouvements. La peau possède également un comportement plastique au cours du temps [SSFD02] d'où la création des rides. Le comportement visco-elastique dans le cas simple linéaire de petites déformations isotropes peut être modélisé correctement par un système de ressorts amortis [Dul04].

On rappelle que pour un comportement élastique linéaire pour de faibles déformations, une action σ entraîne une déformation ϵ tel que $\sigma = E\,\epsilon$, avec E le module d'Young du matériau considéré. L'amortissement est quant à lui caractérisé par son coefficient μ tel que $\sigma = \eta\,\epsilon_{,t}$, où $\epsilon_{,t} = \frac{\mathrm{d}\epsilon}{\mathrm{d}t}$. On obtient alors l'équation différentielle régissant le mouvement simplifié de cette

structure élastique :

$$\sigma = E\,\epsilon + \eta\,\epsilon_{,t}\ .$$

L'équation peut alors être complexifiée par la suite pour prendre en compte de larges déformations non linéaires ou les inhomogénéités spatiales de la peau. Cela rend également l'équation plus difficile à résoudre.

Cas des Muscles

Les muscles sont à l'origine du mouvement. Ils sont généralement séparés en deux catégories : muscles lisses et squelettiques. Les muscles lisses sont responsables de tâches physiologiques se situant sur les bords des organes creux (déglutition, mouvement des yeux, estomac, ...)[1], alors que les muscles squelettiques sont reliés aux os et permettent le mouvement. Ce sont donc ces derniers qui nous intéressent dans le cas d'animation de personnages. Le corps humain en contient environ 700. On peut encore séparer ces muscles squelettiques en deux catégories [Gol91, Ned98], isotoniques et isométriques. Dans le cas isométrique, le muscle se tend sans changer de longueur, ces muscles créent des forces mais n'influent pas sur le mouvement du sujet. Dans le cas isotonique, le muscle change de longueur et de forme ce qui induit généralement un mouvement du squelette. Au niveau anatomique, il existe encore différents types de contractions, celles qui sont maximales à l'extension (mollet) et celles étant maximales en contraction (biceps).

Ces muscles sont constitués par des faisceaux de fibres musculaires ayant une orientation spécifique le long de la contraction qui peut maintenant être étudiée in-vivo par les méthodes d'IRM de diffusion [DKB+99]. Leur arrangement spatial peut être complexe et est responsable de leur comportement anisotrope. Afin de générer le mouvement de contraction, les fibres musculaires d'une dizaine de micromètres de rayon et d'une centaine de long se contractent à volume constant le long de leurs axes propres. La déformation du muscle est alors qualifiée d'incompressible. Enfin, ces muscles sont liés au squelette par les ligaments qui se déforment peu lors de la contraction.

2.1.2 Méthodes Physiques

Les méthodes modélisant la déformation de la peau basées sur la dynamique du mouvement doivent prendre en compte l'aspect élastique des muscles, leur contrainte de volume constant ainsi que l'orientation de leurs fibres pour les méthodes les plus précises. Elles tentent de modéliser les déformations de ceux-ci de façon fidèle afin d'avoir un mouvement induit sous la peau réaliste. Il existe deux grandes méthodes de modélisation des muscles permettant de résoudre les équations différentielles régissant le mouvement. Les méthodes de masses-ressorts amortis et les méthodes d'éléments finis.

Ces deux méthodes consistent à discrétiser les équations du mouvement des muscles. Celle des masses-ressorts est relativement facile à mettre en oeuvre et à résoudre. Cependant elle est extrêmement sensible à la topologie de la discrétisation du volume du muscle et possède notamment des problèmes de convergence suivant les méthodes de résolution. Des variantes permettent par contre de prendre en compte de façon rapide l'anisotropie du volume donné par une orientation fibrée de même que la contrainte de déformation à volume constant [BC00].

La méthode des éléments finis [OMO+03, ZL05, Che92] s'est très développée récemment dans les domaines de la modélisation de structures déformables. Elle donne des résultats pouvant être très précis. Son défaut majeur étant une résolution très longue car les systèmes comportent rapidement des dizaines de milliers d'inconnues.

[1] Le coeur étant pris à part

Equations des déformations

On notera tout d'abord qu'un résumé de certaines équations physiques de base à résoudre dans ces modèles est réalisé en [NMK+06].

Les équations régissant le mouvement du muscle peuvent être plus ou moins complexes suivant le degré de réalisme souhaité. Ainsi, pour animer un muscle, le plus simple est de considérer la déformation d'un point de vue lagrangien. Ainsi chaque position du volume du muscle est suivie au cours de la déformation. On considère un référentiel de repos en l'absence de tout stress $\mathcal{R}^0$. Les positions du muscle dans ce référentiel sont alors désignées par le vecteur $\mathbf{X}$. Après la déformation à un instant t, on considère le référentiel $\mathcal{R}$ tel que les mêmes positions désignant le muscle soient données par $\mathbf{x}$. On appel tenseur de déformation le tenseur tel que ses coefficients soient donnés par

$$\mathrm{F}_{ij} = \frac{\partial x^i}{\partial X^j} \ ,$$

on a alors la relation différentielle sur les lignes infinitésimales $\mathrm{d}\mathbf{x} = \mathrm{F}\,\mathrm{d}\mathbf{X}$. Cette matrice est caractéristique de la déformation subie par le muscle.

Le second tenseur de stress de Piola Kirchhoff est donné par

$$\mathrm{S} = 2\frac{\mathrm{d}W}{\mathrm{dC}} \ ,$$

où W est une énergie de déformation caractéristique du matériaux et C est le tenseur de Cauchy-Green de la déformation donné par $\mathrm{F}\,\mathrm{F}^T$. On peut alors montrer que le tenseur de stress σ est donné par

$$\sigma = \frac{1}{|\mathrm{F}|}\,\mathrm{FSF}^T \ .$$

C'est à partir de l'énergie de déformation W que l'on prend en compte la contraction des fibres musculaires le long de leurs axes. Différents modèles existent pour modéliser les muscles, et il s'agit souvent de relations empiriques.

En notant $\mathbf{b}$ les forces de volumes on peut alors montrer que le problème revient à résoudre le bilan des forces

$$\nabla\sigma + \rho\,\mathbf{b} = 0$$

sur le volume du muscle avec les conditions frontières données par l'application de forces de surfaces ou de fixation du muscle sur les bords. ρ représente ici la densité du muscle.

2.1.3 Cas des méthodes non physiques classiques

Pour des raisons évidentes de rapidité de calcul, les méthodes physiques sont rapidement limitées pour des utilisations temps réel telles que dans les jeux vidéo. Dans ce cas, on utilise différents moyens ne prenant pas en compte le calcul des forces (ou de façon très sommaire). On s'intéresse alors à modéliser les déplacements et les déformations de la surface de l'objet sans se soucier des causes de ceux-ci. La méthode possède évidemment l'inconvénient de ne pas forcément être liée à la réalité. Cependant, il s'agit également d'un avantage pour laisser une part plus importante de liberté à l'artiste qui peut alors créer tout type de déformations.

Il existe trois grandes méthodes de base classiquement utilisées dans le cas de ces déformations de personnages animés [CH01]. Celles-ci sont généralement implémentées dans les logiciels d'animations commerciaux.

Interpolation de Formes

L'interpolation de formes (shape-interpolation ou multi-target morphing ou blend-shape interpolation) stocke le maillage d'un certain nombre de poses caractéristiques. On note par $\mathcal{B}_i$ l'ensemble des sommets de la position i. Pour $N+1$ positions ainsi stockées, l'ensemble $(\mathcal{B}_i)_{i\in[\![0,N]\!]}$

forme une base dans l'ensemble des formes ayant cette topologie et une position intermédiaire peut donc être interpolée dans cette base.

$$\mathcal{B} = \sum_{k=0}^{N} a_k \, \mathcal{B}_k \ ,$$

où les a_k sont les coefficients d'interpolation. Un certain nombre de remarques peuvent cependant être faites.

$$\begin{cases} \mathcal{B} = \sum_{k=0}^{N} a_k \, \mathcal{B}_k \\ \forall a_k \geq 0 \\ \sum_{k=0}^{N} a_k = 1 \ . \end{cases}$$

On peut facilement implémenter une animation en suivant une trajectoire dans l'espace barycentrique des $\mathcal{B}_i$ en faisant dépendre les valeurs de l'interpolation d'un paramètre : $\mathcal{B}(t) = \sum_{k=0}^{N} a_k(t) \, \mathcal{B}_k$. La méthode est souvent utilisée pour l'animation faciale [CB02, Chu05] où l'artiste peut créer à sa guise des expressions caractéristiques fixes (rire, grimaces, ...), puis on réalise le passage de l'une à l'autre par ce type d'interpolation. Un exemple de poses de base est ainsi montré en Fig. 2.1.

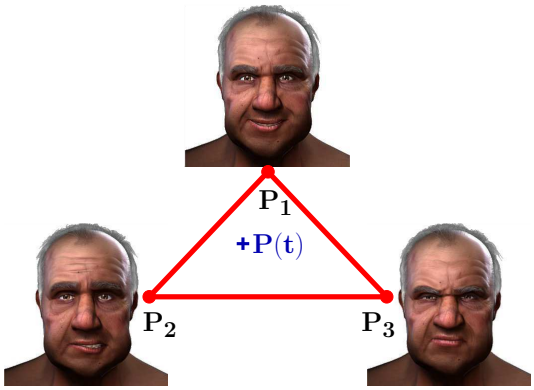


Fig. 2.1: Exemple de morphing multi-cibles dans le cas d'animations faciales. La position intermédiaire doit être interpolée entre les trois poses de bases. Les surfaces du visage proviennent de cas d'exemples issues du logiciel commercial MAYA.

La méthode possède cependant un certain nombre de limitations :

- Elle nécessite une place mémoire importante pour stocker l'ensemble des positions de tous les sommets.
- On ne peut pas créer d'autres animations que celles découlant directement des formes de base.
- L'interpolation linéaire barycentrique simple n'est pas forcément cohérente d'un point de vue physique : deux poses peuvent s'annuler si la somme vectorielle s'annule au cours de la trajectoire dans l'espace des formes. Il est donc difficile de créer des poses qui n'interagissent pas les unes avec les autres.
- La forme interpolée n'a pas le même volume que les formes de base. Ainsi, même si chacune des formes extrêmes $\mathcal{B}_i$ ont toutes le même volume, celle qui est interpolée n'aura pas forcément ce volume.
  On illustre ce phénomène en prenant un cas très simple de deux formes pour un parallélépipède rectangle. La première forme possède des côtés de longueur $a \times b \times c$ alors que le second possède des côtés de longueur $b \times a \times c$. Ces deux formes de base ont évidemment le même volume $V_0 = a\,b\,c$. Une interpolation barycentrique simple donne alors une forme illustrée en Fig. 2.2 vue en projection (le problème étant le même en 2D avec l'aire). Dans ce cas d'interpolation pour $t \in [0, 1]$, le premier côté possède alors une

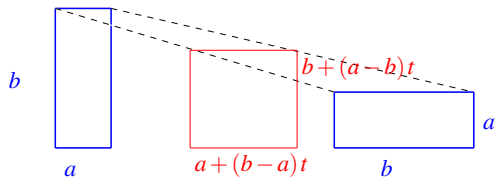


Fig. 2.2: Interpolation barycentrique simple entre deux parallélépipèdes rectangles vues en projection. Le premier parallélépipède possède des longueurs $a \times b \times c$ alors que le second $b \times a \times c$. La forme rouge correspond à celle qui est interpolée linéairement.

longueur $a + (b - a)\, t$ alors que le second est de longueur $b + (a - b)\, t$. Le volume à l'instant t est alors donné par

$$V(t) = abc + c(a - b)^2\, t(1 - t)\ .$$

Le volume n'est donc pas conservé lors de l'interpolation ce que montre de façon évidente la figure. Il passe dans ce cas par un maximum en $t = 1/2$. Et la variation est alors de l'ordre du carré de la différence entre les longueurs des côtés.

Déformation de formes libres

La déformation de formes libres FFD (free form deformation) consiste à déformer non plus directement les sommets de l'objet mais à appliquer une fonction de déformation sur l'espace dans lequel se trouve la forme [Bar84, SP86, Coq90]. L'interpolation des déplacements pour chaque sommet se réalise alors automatiquement car la déformation est continue. On considère généralement que la déformation est réalisée par un volume tridimensionnel de Bézier dont on contrôle la déformation en déplaçant simplement ces points des contrôles [CR94]. La déformation est alors aisément contrôlée par une grille parallélépipédique formée par les 4^3 points de contrôles si l'on prend des polynômes de degré trois. Un exemple de grille déformée est montré en Fig. 2.3.

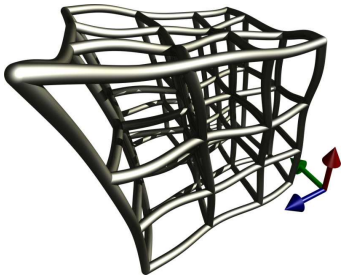


Fig. 2.3: Exemple de grille de base déformée dans laquelle on va placer l'objet. La forme sera alors déformée par application de fonctions d'interpolation entre les points de la grille.

La position $\mathbf{x}(x, y, z)$ déformée du sommet dépendant des points de contrôles $\mathbf{P}_{ijk}$ est alors donnée par

$$\mathbf{x}(x, y, z) = \sum_{(i,j,k)=0}^{3} \Big(\mathbf{P}_{ijk}\, B_i(x) B_j(y) B_k(z)\Big)\ ,$$

où les B_i sont les polynômes de Bernstein de degrés trois par exemple. La méthode possède cependant également des inconvénients qui lui sont propres :

– On ne contrôle pas directement les sommets mais l'espace lui même, il peut être difficile de donner la forme voulue à l'objet.
  Certaines méthodes ne prennent alors plus en compte le volume entourant l'objet mais seulement une approche surfacique avec un unique polygone de contrôle basé directement

sur les sommets de l'objet triangulé [SK00]. Les points de contrôle agissent alors sur les sommets de l'objet déformé par des poids dépendants de leurs distances à ceux-ci. Cela permet un contrôle direct des points de la surface ainsi qu'un aperçu plus intuitif du polygone de contrôle.
Enfin d'autres approches permettent, tout en gardant l'approche volumique de la déformation, de contraindre directement le mouvement à partir des sommets. Pour cela une méthode de moindres carrés permet de déterminer la fonction de déformation [HHK92]. Enfin, plus récemment, une solution explicite à pu être mise en place [HZTS01].

– L'utilisation des polynômes de Bernstein rend la déformation globale à l'intérieur du parallélépipède de contrôle. Cela n'est pas forcément souhaitable, un contrôle local étant le plus souvent plus approprié. Cependant l'utilisation d'autres types de fonctions de base permettent de résoudre ce problème. On peut ainsi penser à l'utilisation des fonctions de Bézier tri-cubiques définies par morceaux [Coq90], ou également aux surfaces splines permettant des jonctions C^2 [GP89, FMP96]. Les déformations sont alors locales. Les fonctions NURBS peuvent également être utilisées pour des déformations plus complexes [CN00, HZTS01].

La méthode peut aussi permettre la déformation de maillages grossiers de façon précise en l'associant à des méthodes de subdivision [FHW98].
La FFD peut également être utilisée pour modéliser des systèmes complexes pour l'animations de personnages comportant plusieurs couches (squelette, muscle et peau). Les muscles étant alors modélisés par FFD, attachés au squelette et peuvent posséder des propriétés élastiques par l'utilisation de systèmes masses-ressorts. Un exemple de système complexe a ainsi pu être réalisé par Chadwick *et al.* [CHP89]. De même, il est possible de lier les FFD à des déformations dynamiques en liant l'espace de contrôle aux forces extérieures subies par l'objet ce qui permet de modéliser des déformations non rigides de façon physique avec un nombre limité de degrés de liberté [FPT97]. Enfin, si la déformation classique ne permet pas de garder un volume constant, il est cependant possible de contraindre celle-ci par une minimisation, et ce, même dans le cas de larges déformations [HML99].

Skinning

Enfin la méthode du skinning ou Skeletal Subspace Deformation (SSD) ou encore vertex blending est probablement la plus répandue au niveau des applications temps réel en jeu vidéo. La déformation des vertex est alors liée directement à la déformation d'un squelette consistant en un ensemble de repères. Les avantages immédiats sont que seule une pose complète de la surface au repos est nécessaire (comme la FFD) et que l'on a un contrôle directe (comme l'interpolation de formes) et rapide sur les repères du squelette. Seul le mouvement du squelette est sauvegardé et donc peu de place mémoire est nécessaire.

C'est principalement cette méthode qui va nous intéresser et nous la détaillons dans la suite.

2.2 Méthode du Skinning

2.2.1 Déformation d'un squelette

On considère initialement qu'un squelette hiérarchique est associé à chaque personnage que l'on possède. Ce squelette correspond donc à un ensemble de repères affines de l'espace que l'on place aux endroits souhaités (généralement aux articulations). Chaque repère peut posséder un parent et des fils. Ce repère hérite alors de la transformation de son père et va donner sa transformation à son/ses fils. On note $\mathcal{B}_k = (\mathbf{e}_i^k)_{i\in[\![1,3]\!]}$ la base du niveau k correspondant au repère associé. La transformation du premier niveau donné par $\mathcal{B}_0$ peut être écrite sous la forme

$$\mathbf{e}_i^0(t) = \mathrm{R}^0(t)\,\underline{\mathbf{e}_i^0}\ , \tag{2.1}$$

où t correspond au paramètre temporel lors de l'animation. Et $\underline{\mathbf{e}_i^0}$ correspond à une base de départ que l'on suppose connue. $\mathrm{R}^0(t)$ est une matrice de transformation orthogonale correspondant généralement à une rotation.

De par la hiérarchie mise en place au niveau k, la transformation matricielle se déduit par produit :

$$\mathbf{e}_i^k(t) = \mathrm{M}(t)\,\underline{\mathbf{e}_i^0} = \left(\prod_{m=0}^{k} \mathrm{R}^m(t)\right)\,\underline{\mathbf{e}_i^0}\,. \tag{2.2}$$

Les matrices de rotation sont paramétrées par un axe de rotation $\mathbf{u}$ et un angle θ ce qui permet également l'utilisation directe des quaternions qui faciliteront l'étape d'interpolation [LPJ01]. L'interpolation se réalise alors par la classique interpolation sphérique Slerp (Spherical Linear Interpolation) [Sho85, BF01]. D'autres méthodes d'interpolation permettent cependant de contourner certains problèmes cités par la suite [KZ05, KCS06, KCZS07, And05].

Au niveau des sommets, la translation pourra, par la suite, également être mise en place en considérant des matrices 4×4 en coordonnées homogènes. En notant M^m ces matrices de transformation du repère local du niveau k, et D, la matrice de changement de repère local vers global d'un joint donné, on peut alors exprimer la transformation du sommet $\mathbf{x}_s$ du niveau k par

$$\begin{aligned} \mathbf{x}_s^k(t) = &\ \mathrm{T}(t)\,\underline{\mathbf{x}_s^0} \\ = &\ \left(\prod_{m=0}^{k} \mathrm{M}^m(t)\right)\,\mathrm{D}^{-1}\underline{\mathbf{x}_s^0}\,, \end{aligned} \tag{2.3}$$

où $\underline{\mathbf{x}}_s$ correspond à la position initiale du sommet. Cette formule est tout simplement l'application d'un changement de repère et d'une rotation exprimée dans le repère local que l'on itère k fois. Un exemple de squelette hiérarchique à trois niveaux est montré en Fig. 2.4.

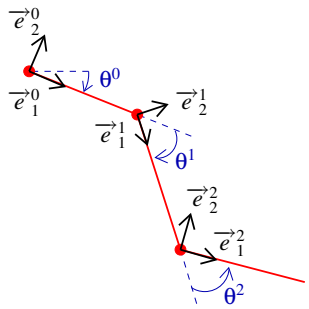


Fig. 2.4: Exemple de hiérarchie d'un squelette à trois niveaux. Les joints sont signalés par les points rouges.

Elle permet donc de mettre en place l'animation de la structure rigide qu'est le squelette d'animation. Dans la suite, nous allons maintenant nous intéresser plus particulièrement à la déformation des sommets formant la peau du personnage en suivant le mouvement de ce squelette. Un exemple de personnage complet avec un squelette d'animation associé est montré en Fig. 2.5.

2.2.2 Assemblage de blocs

La première façon de considérer une animation de personnages graphiques consiste à construire ceux-ci par un assemblage de formes simples (cylindres, cubes, sphères, ...). Les premiers personnages 3D des jeux vidéo étaient ainsi formés d'assemblage de blocs séparés les uns des autre. On peut citer notamment le très connu Tomb Raider, ©*Eidos* de 1996 comme montré en exemple en Fig. 2.9. Ainsi les sommets de chaque bloc ne dépendaient que de la base associée à son repère. Chaque sommet $\mathbf{x}_k$ est alors donné directement par l'équation 2.3.

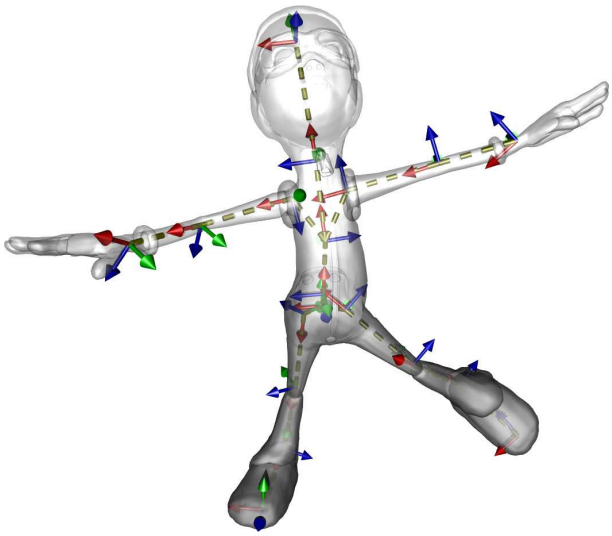


Fig. 2.5: Exemple de personnage complet associé à son squelette d'animtation.

Un exemple est montré pour un bras articulé très simple formé par deux repères. L'un des bras est fixe alors que le suivant subit une rotation autour de la jointure. L'animation du squelette est montrée pour trois angles en Fig. 2.6.

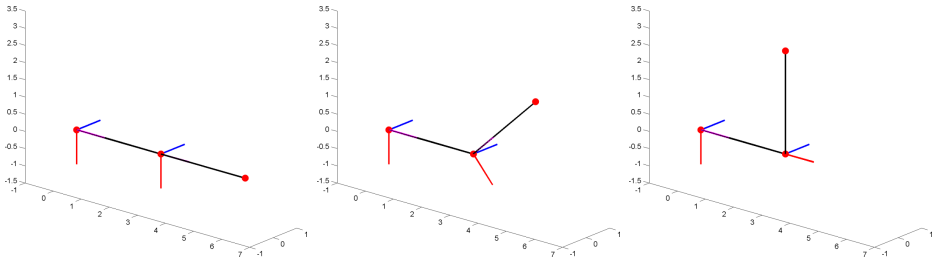


Fig. 2.6: Squelette du système montré pour les angles $\theta = 0$, $\frac{\pi}{4}$ et $\frac{\pi}{2}$.

En plaçant autour de ces axes un cylindre simple et en déplacant les sommets suivant l'éq. 2.3, on obtient le résultat montré en Fig. 2.7 On peut constater que l'on a un évident problème

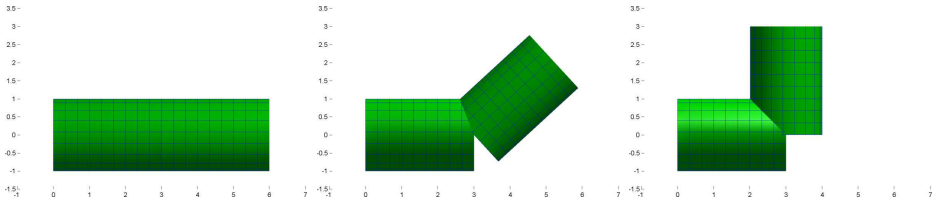


Fig. 2.7: Déformation des sommets des deux cylindres par blocs.

de continuité à la jonction. En effet, de par la rotation, les blocs vont laisser des trous. Une première solution consistait à essayer de combler ces trous en ajoutant des primitives telles que des sphères à l'emplacement du pivot afin de ne pas visualiser les bords. Cette méthode fonctionne correctement lorsque les couleurs sont unies, par contre il est compliqué d'appliquer des textures à ces jointures. Ainsi, dans les jeux utilisant ce type de techniques, il était courant d'avoir une texture uniforme (sans effet diffus et spéculaire lié à la normale) aux jonctions.

2.2.3 Skinning simple

Une première amélioration va consister à abandonner la construction de l'objet uniquement par des primitives simples, mais par une vraie surface pouvant représenter la peau. Pour cela, la méthode suivante a d'abord été désignée par le terme de skinning. Pour ne plus différencier typiquement deux blocs cylindriques, on modifie maintenant la connectivité du maillage formé par la surface de façon à relier les deux cylindres. On obtient alors une unique surface en terme de connectivité. Celle-ci n'est plus entièrement décrite par les primitives de bases et devient donc une surface triangulée quelconque.

L'exemple de l'animation est alors repris en Fig. 2.8 après avoir connecté les deux cylindres.

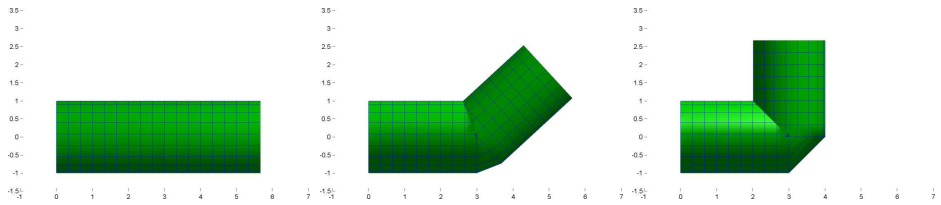


Fig. 2.8: Déformation des vertex des deux cylindres en une seule fois après avoir lié les deux primitives.

La méthode a été largement utilisée dans les jeux vidéo comme le montre la Fig. 2.9 car il n'y a plus de problèmes de trous aux articulations [And01]. Par contre, le coude de la jointure subit évidemment une déformation très importante car les triangles (ou quads) deviennent très étirés. L'application de textures sur les coudes est donc encore problématique. De plus on a toujours une interpénétration de la surface de l'autre côté du coude même pour des angles faibles ce qui rompt la continuité lors de l'application de textures. Enfin, la déformation n'est pas lisse car la surface n'est pas C^1 sur les jointures ce qui pose des problèmes de discontinuités lors de l'éclairage des bords.

Fig. 2.9: Exemple d'un personnage de jeu vidéo créé par ©*Eidos* dont l'animation est créée par la méthode présentée. La première image correspond au premier titre de la série de jeu et le personnage est entièrement modélisé par des assemblages de blocs non connectés. La seconde image quant à elle provient du troisième titre de la série. Cette fois, le personnage ne forme plus qu'une unique surface.

2.2.4 Soft skinning

Afin de régler en partie ces problèmes et d'avoir une déformation lisse, il est nécessaire de modifier la position des sommets [And01, LPJ01, PH02]. Ainsi, la méthode de soft skinning va maintenant considérer que chaque sommet n'est plus lié à un repère unique mais à plusieurs.

Un point sur la jointure de l'exemple du bras n'a en effet pas de raison d'appartenir plus à une partie du squelette qu'à l'autre. On va donc faire dépendre ces sommets de leurs repères voisins avec des variations continues de façon à obtenir une déformation lisse comme le montre la Fig. 2.10 pour le cas d'un cylindre. Cette méthode est notamment utilisée dans plusieurs jeux

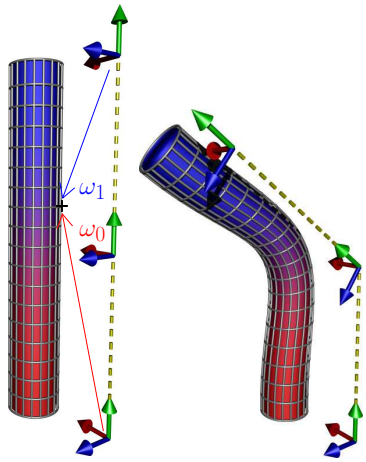


Fig. 2.10: Exemple de cylindre déformé en choisissant des sommets dépendant du premier et du dernier joint. Les sommets les plus bas dépendent principalement du joint du bas, alors que les sommets les plus élevés dépendent du joint supérieur. Les poids ω_0 et ω_1 caractérisent la contribution de l'orientation de chaque joint sur un sommet donné. La figure de droite montre la déformation obtenue lorsque l'on a effectué une rotation de $\frac{\pi}{4}$ de l'orientation du joint central.

vidéos actuels. Un exemple est ainsi montré en Fig. 2.11

Fig. 2.11: Exemple d'un personnage de jeu vidéo créé par ©*Eidos*. Ce personnage est maintenant tiré du septième titre de la série du jeu ; ie est entièrement texturé avec des jonctions lisses.

Pour cela, on va en théorie supposer que tout sommet peut dépendre de chaque repère de l'objet. En supposant que le sommet s dépend de N_r repères, on a alors la relation [LPJ01]

$$\mathbf{x}_s(t) = \left(\sum_{i=1}^{N_r} \omega_i^s \, \mathrm{T}_i(t) \right) \underline{\mathbf{x}_s^0} \, , \tag{2.4}$$

où ω_i^s correspond au poids donné pour le sommet s par rapport au repère i. Ces poids sont indépendants de t et donc fixés sur le squelette au repos. En notant que chaque repère i est à la

position hiérarchique k_i, on peut alors développer l'expression avec les rotations

$$\mathbf{x}_s(t) = \left[\sum_{i=1}^{N_r} \omega_i^s \left(\prod_{m=0}^{k_i} \mathrm{M}^m(t) \right) \left(\mathrm{D}^i\right)^{-1} \right] \underline{\mathbf{x}_s^0} \ . \tag{2.5}$$

Afin de ne pas créer de déformations involontaires de l'objet, les poids ω_i^s sont généralement considérés de façon barycentrique tels que pour tout sommet s

$$\begin{cases} \sum_i \omega_i^s = 1 \\ \\ \forall i \ , \ \ \omega_i^s \geq 0 \ . \end{cases}$$

On remarquera enfin que la matrice de transformation des joints se construit de façon itérative le long du parcours du squelette et ne nécessite donc qu'un seul parcours à chaque intervalle de temps.

La continuité de la courbe est grandement améliorée et l'application de texture sur l'ensemble de la surface est désormais possible. Un exemple pour le cas des morceaux de cylindre est ainsi montré en Fig. 2.12. De plus, il est facile de mettre également cette déformation en place sur les normales de la surface avec la relation

$$\mathbf{n}_s(t) = \left[\sum_{i=1}^{N_n} \omega_i^s \, \mathrm{T}_{i|_R} \right] \underline{\mathbf{n}_s^0} \ , \tag{2.6}$$

où $\mathrm{T}_{i|_R}$ correspond à la matrice de rotation 3×3 associée à la matrice T_i. $\mathbf{n}_s$ représente la normale associée au sommet $\mathbf{x}_s$, et $\underline{\mathbf{n}_s^0}$ est celle de la forme de départ.

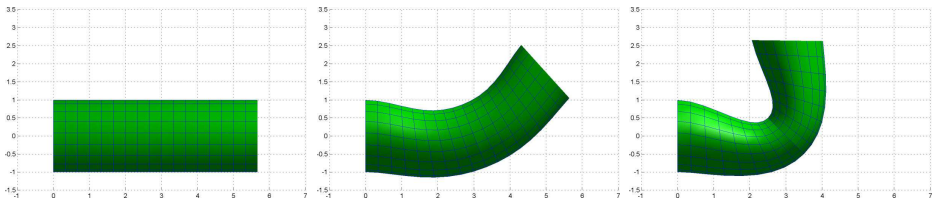


Fig. 2.12: Déformation des vertex par la méthode du smooth skinning en prenant des poids dépendant du carré de la distance au segment des squelettes.

On notera cependant que dans le cas du cylindre pris en exemple, on peut observer une nette réduction du volume lorsque l'articulation se plie.

2.2.5 Difficultés, Choix des poids

On peut maintenant comprendre que la principale difficulté consiste à trouver les poids ω_i pour chaque sommet rendant le meilleur résultat. Ces poids doivent généralement être déterminés à la main pour obtenir de bons résultats. Cependant des méthodes automatiques permettent d'alléger une partie du travail. Une façon généralement utilisée consiste à donner à ω_i^s une valeur proportionnelle à l'inverse du carré de la distance du sommet s au segment du squelette lié au repère i [Blo02, Rou04]. Une fois les distances calculées, les poids peuvent être normés afin d'obtenir une somme unitaire.

En utilisant cette méthode de choix de poids, on obtient alors les résultats du type montré en Fig. 2.12 et 2.10.

On remarque cependant que pour des angles de déformation importants, on obtient un problème de réduction du rayon des cylindres. Ce phénomène est bien connu sous le nom de "collapsing elbow" car il intervient notamment lors de la modélisation du bras [LCF00, YZ05].

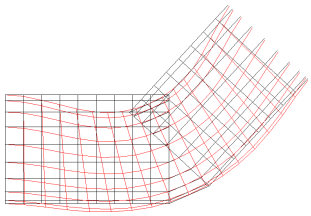


Fig. 2.13: Comparaison entre la surface obtenue par simple accolement des deux cylindres (noir) et celle obtenue par soft skinning (rouge)

On montre en Fig. 2.13 la comparaison de la surface obtenue par la méthode de soft skinning et celle du skinning simple sans déformations.

On montre également que cet effet est obtenu sur des surfaces plus complexes de personnages complets en Fig. 2.14.

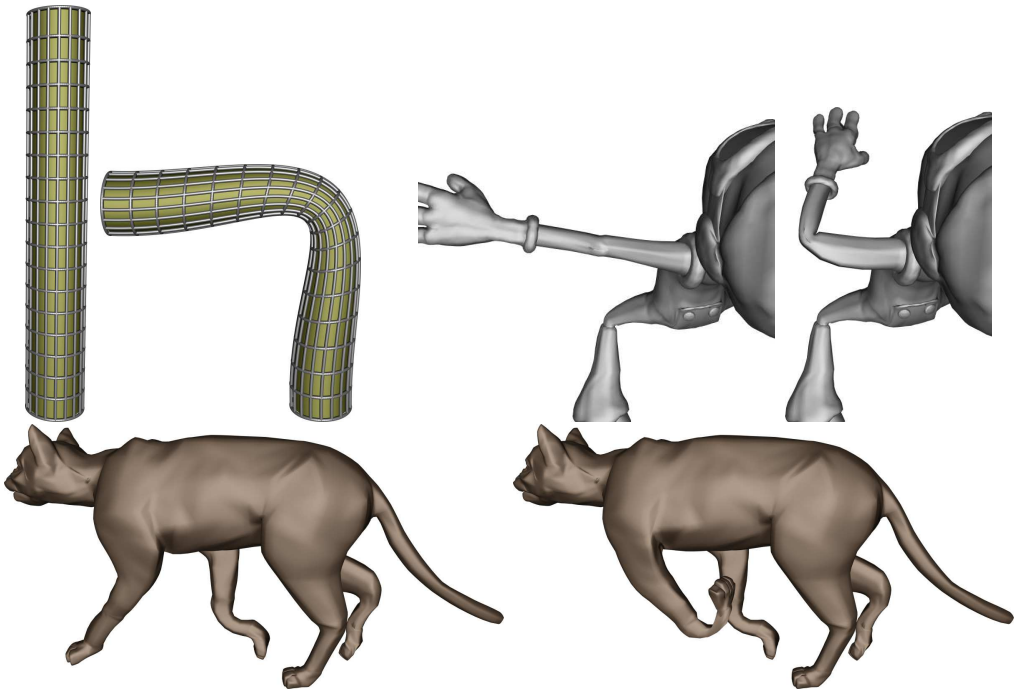


Fig. 2.14: Exemple de l'effet de "collapsing elbow". Les deux premières images correspondent à cet effet visualisé sur le modèle du cylindre dans le cas d'une rotation de $\frac{\pi}{2}$. On peut également visualiser ce problème dans le cas de surfaces plus complexes. Ainsi les deux images suivantes correspondent au personnage vu de haut qui plie le bras. On peut y constater une déformation importante de la taille de son bras. De même l'exemple du chat montre le cas d'une flexion importante d'une patte qui perd de façon très visible une grande partie de son volume.

2.3 Solutions existantes pour le skinning

Du à ce problème de déformation pour des angles élevés, différentes méthodes ont été présentées afin de réduire cet effet :

2.3.1 Modification des poids et limitation en angle

Yang et Zhang [YZ05] proposent de limiter les problèmes typiques du coude en faisant notamment dépendre les poids de skinning d'une expression polynomiale dépendant des angles

entre les joints. De plus, ils tentent d'empêcher le problème d'interpénétration en définissant un angle limite lors de la rotation de ce coude. Ils s'intéressent également à la perte de volume lors de la déformation et tentent de minimiser celle-ci en modifiant la forme de la surface pour mimer de façon géométrique le positionnement des parties graisseuses. Cette modification n'est cependant pas exacte et est trouvée de façon géométrique empiriquement. La technique fonctionne correctement pour le cas typique du "collapsing elbow" ; cependant, les auteurs admettent que cette méthode est moins adaptée pour les cas du corps où les joints peuvent être liés à plus de deux os.

2.3.2 Ajout de repères

Une autre façon simple de limiter ces problèmes de déformations pour des angles importants consiste à ajouter des repères fictifs à certains endroits. Cela permet de diminuer l'écart angulaire entre deux repères et de lisser la déformation d'une façon plus contrôlable. Mohr et Gleicher [MG03] développent une façon automatique d'ajouter ces nouveaux repères de façon à réduire par deux les écarts angulaires des joints lors de l'interpolation. De par l'augmentation du nombre de joints, il est cependant nécessaire d'avoir à disposition un plus grand nombre de poids.
Pour cela, on résout à nouveau un système linéaire surdéterminé dans un sens de moindres carrés sur un ensemble d'exemples d'apprentissage qu'il est nécessaire de posséder.

2.3.3 Méthode multi-poids

Une autre méthode permettant de gérer le problème du "collapsing elbow" ou "candy wrapper" est d'augmenter les degrés de liberté de la déformation en ne définissant non plus seulement un seul poids par jonction mais plusieurs. Pour chaque coefficient de la matrice de transformation associée au repère correspondant, on définit un poids différent. Cette méthode de skinning développée par Wang et Phillips [WP02] peut alors s'écrire, en reprenant l'éq. 2.4, par la relation matricielle

$$\mathbf{x}_s(t) = \left[\sum_{i=1}^{N_r} \Omega_i^s \, \mathrm{T}_i(t) \right] \underline{\mathbf{x}_s^0} \ ,$$

où cette fois, Ω_i^s est une matrice 4×4. Cette méthode peut se révéler puissante car elle permet une grande liberté au niveau de la déformation. Cependant, elle est également peu intuitive et si un travail conséquent est d'ores et déjà nécessaire pour l'artiste pour déterminer un unique poids par repère pour chaque sommet, il est peu concevable de pouvoir maîtriser avec précision l'influence de chaque poids lorsque l'on en considère plusieurs. Wang et Phillips utilisent alors une méthode de moindres carrés afin de déterminer ces poids de façon automatique en ayant préconçu des poses de l'animation desquelles on souhaite se rapprocher.

2.3.4 EigenSkin

Kry *et al.* [KJP02] proposent une autre façon d'utiliser des poses précalculées pour le skinning. Pour cela, on compare l'erreur obtenue entre la position donnée par le skinning classique et celles des poses précalculées déformées. On se définit pour cela une pose de comparaison correspondant à celle de repos. L'erreur est alors calculée pour chaque sommet de l'objet. Une correction exacte à appliquer sur les sommets de la pose initiale peut alors être connue afin de pouvoir la déformer par un skinning classique. Pour éviter d'avoir à mémoriser l'ensemble des corrections pour chaque pose, on garde seulement les composantes principales des vecteurs de correction à l'aide d'une décomposition en valeurs singulières. Les auteurs préconisent de garder deux à cinq vecteurs propres pour obtenir des résultats convaincants.

2.3.5 Axe médian

En reprenant un skinning classique, Bloomenthal [Blo02] propose de baser la déformation non plus seulement sur le squelette d'animation mais sur une surface (et morceaux de droites) également appelé médiane. Cette surface est définie comme la réunion des centres des boules (sphères) maximales contenues dans l'objet [FLM03, ABE07]. Une fois cette surface définie sous forme triangulée, la transformation de skinning définie jusqu'à présent est appliquée sur celle-ci. On utilise ensuite le fait que cette surface fournit le champ des distances au squelette autour duquel se situe la peau. On peut alors reconstruire la peau par une simple convolution d'un filtre tridimensionnel le long de la surface médiane [BS91]. Cela revient alors à faire rouler une sphère sur la surface dont le rayon dépend de la valeur de l'axe médian. La surface correspond alors à l'ensemble des positions parcourues par l'extrémité de cette sphère.
Cette méthode permet de résoudre le problème de creux lors d'une rotation d'angle important, de même que le problème classique du "candy wrapper" qui a lieu lorsque l'on tord la jonction à 180°. Par contre, elle a l'inconvénient de nécessiter le précalcul de cette surface médiane qui peut se révéler long et surtout instable notamment pour le calcul numérique. En effet, cette surface n'est pas continue en fonction de la position des sommets. Enfin la méthode de reconstruction par convolution ne permet pas de gérer le problème de contrainte de volume.

2.3.6 Construction d'un squelette flexible

Finalement James et Twigg [JT05] proposent quant à eux une solution originale où seul le maillage des poses typiques de l'animation est disponible. Le squelette n'est alors plus nécessaire. Celui-ci est déterminé de façon automatique par une méthode de moindres carrés. Le squelette ainsi trouvé n'est cependant pas comparable au squelette rigide utilisé. Celui-ci peut se déformer de façon affine quelconque le long de l'animation. Les auteurs qualifient donc le squelette ainsi trouvé d'os flexibles. Les poids sont par la suite également déterminés de façon automatique en utilisant une méthode de moindres carrés non négatifs (NNLS).
Encore une fois, des poses d'apprentissage sont toujours nécessaires.

2.3.7 Approche choisie

Chaque méthode proposée possède ses avantages et inconvénients. Certaines sont assez générales (axe médian) mais sont parfois lourdes à mettre en place, d'autres nécessitent la mise en place de poses au niveau du maillage ce qui demande un travail supplémentaire à l'artiste. D'autres méthodes sont juste basées sur des cas particuliers et ont pour but de ne fonctionner que dans ces circonstances.

Notre approche consiste à prendre en compte un skinning réalisé par des interpolations linéaires classiques sans demander un travail supplémentaire à l'artiste. Les méthodes les plus robustes et pouvant s'appliquer dans les cas les plus généraux sont généralement celles découlant de la physique. Or nous avons pu constater que la déformation par skinning classique ne conservait pas un volume constant lors de l'animation. Cette propriété physique est pourtant présente pour les muscles, parties adipeuses et fluides (sang) des personnages animés. Nous souhaitons donc mettre en place une méthode de conservation de volume de façon générale sur une déformation classique. Cela nous permettra de nous rapprocher de la relation d'incompressibilité liée à la physique des tissus constituant nos personnages.

Chapitre 3

Contrainte de volume

3.1 Expression d'un domaine bordé par une surface fermée

3.1.1 Cas général

Nous allons maintenant nous intéresser tout d'abord au volume global de l'objet. Pour cela, on suppose que celui-ci est décrit paramétriquement par $\mathbf{x}(u,v) = (x(u,v), y(u,v), z(u,v))$, avec $(u,v) \in \mathcal{D}$, où $\mathcal{D}$ est le domaine de définition de la surface. Cet objet peut alors être caractérisé par ses moments d'ordres quelconques. Le moment d'ordre zéro notamment fournit le volume (voir Fig. 3.1) :

$$V = \int_\Omega \mathrm{d}\Omega \ ,$$

où Ω représente le domaine bordé par les contours de l'objet (Ω doit être un compact de $\mathbb{R}^3$ donc les bords sont au moins C^1 par morceaux). On rappelle ensuite le théorème de la divergence (ou théorème de Green-Ostrogradski) sur une fonction vectorielle $\mathbf{f}$ quelconque et C^1 sur le domaine Ω :

$$\int_\Omega \nabla \cdot \mathbf{f} \mathrm{d}\Omega = \int_{\partial\Omega} \mathbf{f} \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|} \, \mathrm{d}(\partial\Omega) \ , \tag{3.1}$$

où $\mathbf{n}$ correspond à la normale de la frontière en cette position. On considère maintenant [GCP97] $\mathbf{f}(x,y,z) = z\,\mathbf{u}^z$, où $\mathbf{u}^z$ définit un vecteur unitaire dirigé suivant l'axe z. On remarque que l'on

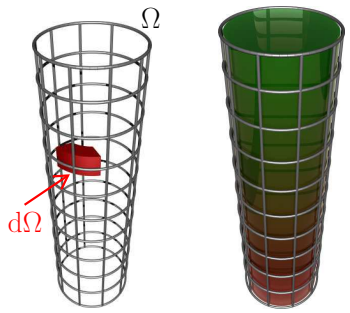


Fig. 3.1: Illustration du calcul d'un domaine bordé par une surface fermée correspondant ici au cas du cylindre. La première figure illustre le calcul du volume par intégration des volumes élémentaires. La seconde figure illustre quant à elle l'exemple d'une fonction $\mathbf{f}$ variant uniquement suivant l'axe z.

a évidemment $\nabla \cdot \mathbf{f}(x,y,z) = 1$ sur Ω. On peut alors écrire

$$\int_\Omega \mathrm{d}\Omega = \int_{\mathcal{D}} z(u,v)\, n^z(u,v)\, \mathrm{d}u\mathrm{d}v \ ,$$

où n^z désigne la composante normale suivant z au point considéré.

3.1.2 Cas triangulé

Dans notre cas, nous considérons généralement une surface triangulée. Cette intégrale se décompose donc sur chaque triangle de la surface. En notant T_i un triangle de l'ensemble $\mathcal{T}$ des triangles composant la surface et en utilisant la linéarité de l'intégrale

$$V = \sum_{T_i \in \mathcal{T}} \int_{\mathbf{x} \in T_i} z_{T_i}(u,v)\, n^z_{T_i}(u,v)\, \mathrm{d}u\, \mathrm{d}v \ .$$

On peut cependant remarquer que la normale de chaque triangle est une constante d'où l'expression

$$V = \sum_{T_i \in \mathcal{T}} n^z_{T_i} \int_{\mathbf{x} \in T_i} z_{T_i}(u,v)\, \mathrm{d}u\, \mathrm{d}v \ . \tag{3.2}$$

On illustre cette expression par la Fig. 3.2. Considérons la surface donnée par le triangle T_i

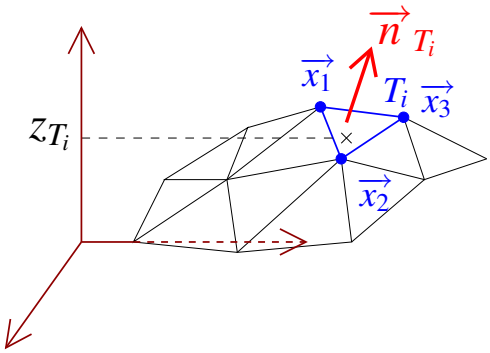


Fig. 3.2: Illustration de l'intégrale de volume sur un triangle.

formé par les trois sommets $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$. Un point quelconque $\mathbf{x}$ appartient à cette surface si les vecteurs $\mathbf{x}_2 - \mathbf{x}_1$, $\mathbf{x}_3 - \mathbf{x}_1$ et $\mathbf{x} - \mathbf{x}_1$ sont liés. On obtient donc l'équation de ce plan par

$$\det(\mathbf{x}_2 - \mathbf{x}_1, \mathbf{x}_3 - \mathbf{x}_1, \mathbf{x} - \mathbf{x}_1) = 0 \ ,$$

ce qui peut s'écrire sous la forme

$$\begin{aligned} & \begin{vmatrix} y_2 - y_1 & y_3 - y_1 \\ z_2 - z_1 & z_3 - z_1 \end{vmatrix} (x - x_1) \\ - & \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ z_2 - z_1 & z_3 - z_1 \end{vmatrix} (y - y_1) \\ + & \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix} (z - z_1) = 0 \ . \end{aligned}$$

On obtient donc directement le vecteur normal en identifiant l'équation par l'expression

$$(\mathbf{x} - \mathbf{x}_1) \cdot \mathbf{n} = \mathbf{x}_0 \ ,$$

où $\mathbf{n}$ est un vecteur normal. La composante z de ce vecteur normal pour le triangle T_i est donc donnée par

$$n^z = \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix} \ .$$

D'un autre côté, paramétrons le triangle par des vecteurs de bases $\mathbf{x}_2 - \mathbf{x}_1$ et $\mathbf{x}_3 - \mathbf{x}_1$. L'équation paramétrique de ce triangle dans cette base est alors donnée par

$$\mathbf{x}(u,v) = (\mathbf{x}_2 - \mathbf{x}_1)\, u + (\mathbf{x}_3 - \mathbf{x}_1)\, v + \mathbf{x}_1 \ ,$$

avec $(u,v) \in \mathcal{T}_0$, où $\mathcal{T}_0$ est la surface donnée par le triangle rectangle isocèle en 0 et de longueur unité comme illustré en Fig. 3.3. On peut alors intégrer sa composante z sur l'ensemble du

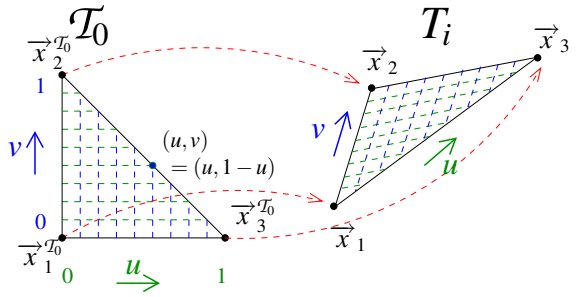


Fig. 3.3: Exemple de paramétrisation des triangles suivant les côtés.

triangle

$$\iint_{(u,v)\in\mathcal{T}_0} z(u,v)\mathrm{d}u\,\mathrm{d}v$$
$$=$$
$$\int_{u\in[0,1]}\int_{v\in[0,1-u]} \Big[(z_2 - z_1)\, u + (z_3 - z_1)\, v + z_1\Big]\mathrm{d}u\,\mathrm{d}v \ .$$

Ce qui fournit, tout calcul fait

$$\iint_{(u,v)\in\mathcal{T}_0} z(u,v)\mathrm{d}u\,\mathrm{d}v = \frac{1}{6}\,(z_1 + z_2 + z_3) \ .$$

En reprenant l'eq. 3.2, le volume du domaine bordé par la surface triangulée peut donc s'exprimer par (voir Fig. 3.4)

$$V = \sum_{(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)\in\mathcal{T}} \frac{1}{6}\,(z_1 + z_2 + z_3) \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix} \ . \tag{3.3}$$

3.1.3 Interprétation

Tout d'abord, on peut exprimer la relation précédente par

$$V = \sum_{\mathcal{T}} z_{\text{avg}}\, \mathcal{A} \ , \tag{3.4}$$

$z_{\text{avg}} = \frac{1}{3}(z_1 + z_2 + z_3)$ étant la moyenne arithmétique des hauteurs z du triangle. Et $\mathcal{A}$ est donné par

$$\mathcal{A} = \frac{1}{2} \begin{vmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{vmatrix}$$

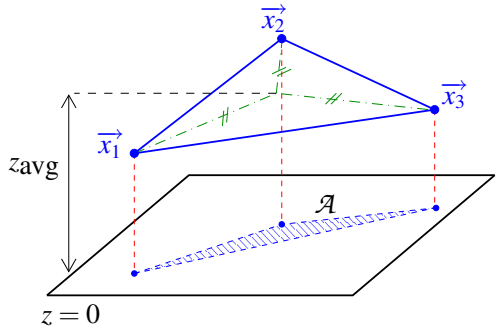


Fig. 3.4: Expression du volume du domaine bordé par la surface fermée triangulée par le produit entre l'aire de la projection du triangle sur le plan $z = 0$ et multiplication par sa hauteur moyenne.

que l'on peut également écrire sous la forme de la composante z du vecteur normal

$$\mathcal{A} = \frac{1}{2}\left(\left(\mathbf{x}^2 - \mathbf{x}^1\right) \times \left(\mathbf{x}^3 - \mathbf{x}^1\right)\right) \cdot \mathbf{e}^z \ ,$$

où $\mathbf{e}_z$ est le vecteur unitaire orienté suivant z. Ce paramètre $\mathcal{A}$ peut se voir également comme l'aire signée du triangle projeté sur le plan $z = 0$ par l'interprétation du déterminant.

Le volume correspond donc à la somme des sous-volumes signés de l'aire des triangles projetés sur le plan (x, y) multipliés par la hauteur moyenne de chaque triangle. On remarquera que cette équation possède l'avantage d'être exacte pour toute surface fermée triangulée.

On pourrait également considérer n'importe quelle autre direction de projection. La généralisation est donc possible en prenant en compte l'aire du triangle projeté sur le plan de normale $\mathbf{n}_p$ et multiplier par la hauteur moyenne dans la direction donnée par $\mathbf{n}_p$. Il suffirait pour cela de reprendre l'éq. 3.1, et de considérer $\mathbf{f} = (\mathbf{x} \cdot \mathbf{n}_p)\ \mathbf{n}_p$. Cela fournit alors l'équation

$$V = \int_{(u,v)\in\mathcal{D}} \mathbf{x}(u, v) \cdot \mathbf{n}_p(u, v)\, \mathrm{d}u\mathrm{d}v$$

Le volume est alors indépendant du choix du plan de projection tant que le surface est topologiquement fermée ce qui donne un sens à la notion de volume.

Enfin, cette définition peut également être utilisée dans le cas de surfaces de subdivision pour lesquelle encore peu d'informations sont accessibles d'un point de vue théorique. Un calcul direct sur ce genre de surface augmente le travail nécessaire de façon exponentielle suivant le raffinement. Une méthode permettant d'avoir une complexité constante au cours des raffinements a cependant été développée dans [PN97].

3.1.4 Utilisation de la trilinéarité

On remarquera finalement que ce volume s'exprime par une équation trilinéaire en x, y et z de par son expression sous forme de déterminant. On peut également noter que cette forme peut se développer une expression polynomiale du type

$$V = \sum_{T\in\mathcal{T}} \sum_{(i,j,k)\in[\![1,3]\!]} \alpha_{ijk}\ x_i y_j z_k \ ,$$

avec $\alpha_{12k} = \alpha_{01k} = \alpha_{20k} = \frac{1}{6}$ et $\alpha_{10k} = \alpha_{02k} = \alpha_{21k} = -\frac{1}{6}$.

On dénote maintenant par abus de notation (i,j,k) les trois sommets du triangle T. On peut alors écrire, pour une surface comportant N_s sommets

$$V = \sum_{(i,j,k)\in[\![1,N_s]\!]} \beta_{ijk}\ x_i y_j z_k \ , \tag{3.5}$$

où $\beta_{ijk} = \sum_{T \in \mathcal{T}} \alpha_{ijk}$ où α reprend les valeurs précédentes dans le cas où (i, j, k) correspond au triangle T et 0 sinon. β correspond donc à un opérateur de voisinage v_1 et est donc très creux. La trilinéarité du volume apparaît ici clairement.

Une notation sous forme de produit scalaire nous sera également utile par la suite. En effet, on peut remarquer d'après cette expression polynomiale que le volume peut s'exprimer sous la forme

$$V =< \mathbf{V}^{YZ}, \mathbf{x} >=< \mathbf{V}^{XZ}, \mathbf{y} >=< \mathbf{V}^{XY}, \mathbf{z} > \tag{3.6}$$

où l'on dénote par $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ les vecteurs des coordonnées des N_s sommets. Et on utilise les notations

$$\begin{cases} V_i^{YZ} = \sum_{(j,k)} \beta_{ijk}\ y_j z_k \\ V_j^{XZ} = \sum_{(i,k)} \beta_{ijk}\ x_i z_k \\ V_k^{XY} = \sum_{(i,j)} \beta_{ijk}\ x_i y_j\ , \end{cases} \tag{3.7}$$

et $< \mathbf{a}, \mathbf{b} >$ correspond au produit scalaire dans $\mathbb{R}^{N_s}$.

3.2 Méthodes existantes de conservation de volume

Nous allons maintenant nous intéresser à certaines méthodes déjà utilisées afin de contraindre un volume constant lors de la déformation. Plusieurs méthodes ont déjà été développées afin de limiter ce problème, cependant aucune n'a encore été appliquée directement pour le skinning.

Certaines méthodes sont liées à celles de la déformation. Nous n'allons pas tenter de présenter de façon exhaustive l'ensemble des méthodes de déformation de maillage. Cependant nous introduirons certaines méthodes récentes qui nous ont semblé intéressantes dans le cadre de la conservation de volume.

Tout d'abord les méthodes de contrôle du volume dépendent du choix de représentation de la surface. Les méthodes de représentation implicites semblent plus faciles à contrôler sur ce domaine car le volume est aisément défini par intégration suivant le signe des valeurs de la fonction implicite. Ces représentations liées au contrôle du volume furent les premières prises en compte.

3.2.1 Surfaces implicites

Contrainte de volume

Desbrun et Cani [DC95, CD97] mettent en place une méthode locale de contrôle de volume pour la déformation de surfaces implicites. La localité de la correction du volume est définie par des régions dans lesquelles certaines particules placées dans le volume vont avoir une action. Ces particules sont placées au centre de ces régions ainsi que sur la frontière de l'objet. Leurs positions sont suivies lors de la déformation au cours du temps et sont utilisées pour calculer les variations du volume de l'objet. Au cours de chaque instant, la fonction est déformée suivant une équation différentielle permettant de limiter le changement de volume.

Cette méthode s'adresse dans ce cas aux surfaces implicites engendrées par des squelettes ponctuels, bien que l'on puisse supposer l'étendre à d'autres méthodes de constructions par dilatation de primitives.

On notera cependant que seules les variations du volume de l'objet sont estimées. La correction étant réalisée par un contrôleur, la méthode est donc sensible aux erreurs numériques. Ce volume total risque donc de diverger lentement au cours du temps.

Fluides incompressibles

Une méthode de suivi de positions appartenant au volume est également utilisée dans le cas de déformations par la méthode de level-set [OS88, OF03]. Cette méthode permet de déformer une surface définie implicitement selon un champ de vecteurs pouvant représenter les forces s'appliquant sur le solide. Cette méthode permet de simuler des déformations liées à la physique (fluide, gaz et solide) mais nécessite la résolution numérique à chaque instant d'une équation aux dérivées partielles qui rend son utilisation lente.

Dans le cas de simulation de fluides incompressibles, on peut alors noter que le champ des vitesses est à divergence nulle ce qui implique que le fluide est déplacé sans variation de volumes et ce, de façon locale (ni extension, ni compression).

La mise en place de l'équation de déformation fait appel à un point de vue eulérien ; cependant, les auteurs remarquent qu'une résolution numérique directe ne permet pas de conserver le volume global du fait de perte de masse lors de déformations importantes (erreurs liées à la discrétisation).

Ils développent ainsi des méthodes semi-Lagrangiennes permettant de traquer certaines positions au cours de la déformation par intégration le long des lignes de flux (Équation Différentielle Ordinaire), et ainsi, de respecter la contrainte d'un volume constant [FF01, EFFM02].

Cependant, encore une fois, le volume peut légèrement varier, du aux imprécisions numériques utilisées pour la discrétisation de l'équation de façon spatiale. Le volume est donc gardé "approximativement" constant, mais sans garantie d'exactitude.

3.2.2 Free Form deformation

Les méthodes précédentes s'adressent aux surfaces implicites, cependant, nous nous intéressons plus particulièrement aux surfaces maillées. Celles-ci sont, de façon générale, moins adaptées aux déformations. Les premières méthodes permettant de lier le volume à la déformation ont été réalisées sur les déformations de formes libres (FFD).

Rappoport *et al.* [RSB96] définissent une déformation de formes libres avec préservation de volume. Leur méthode se base sur la représentation volumique de la forme afin d'en obtenir le volume par intégration dans le cube de contrôle. Afin de déterminer ce volume, ils expriment analytiquement la matrice Jacobienne de la transformation suivant les points de contrôle. La variation de volume par rapport à ces points de contrôle est également exprimée. Enfin une fonctionnelle d'énergie est mise en place afin de limiter la déformation. L'expression considérée étant celle provenant de la théorie des plaques minces [TQ94]. De plus, les auteurs rajoutent une contrainte de continuité afin de pouvoir accoler les différentes primitives après la déformation. La méthode des multiplicateurs de Lagrange [QSS00] est utilisée pour résoudre la minimisation sous contrainte. L'équation non linéaire est ensuite minimisée par la méthode d'Uzawa introduite dans [AHU59]. Leur méthode reste cependant assez lourde à mettre en place. Cependant, les auteurs sont plus axés sur la possibilité de lier leur méthode aux déformations basées sur la physique, ce qui n'est à priori pas notre but premier.

Quelques années après, Hirota *et al.* [HML99] ont présenté également une méthode de déformation de FFD préservant le volume global de la forme. De plus, leur approche est plus adaptée à notre cas car elle ne se base pas sur une approche volumique liée au polygone de contrôle comme dans le cas précédent.

Pour cela, l'utilisateur commence par placer sans restrictions les points de contrôles où il le souhaite. Ensuite, pour l'étape de minimisation, le volume de la forme est exprimé par l'éq. 3.4 de même que sa variation en fonction de la position de chaque sommet $\frac{\partial V}{\partial \mathbf{x}}$. La déformation du volume élastique est quant à elle caractérisée par une fonctionnelle d'énergie utilisée pour caractériser un réseau de masses-ressorts. En notant N_s le nombre de sommets et L_{ij} la longueur

entre les positions $\mathbf{x}_i$ et $\mathbf{x}_j$, on peut alors écrire l'énergie par une expression quadratique classique

$$E_{\text{spring}} = \sum_{i=1}^{N_s} \sum_{j=0}^{N_s} \frac{k_{ij}}{2} \left(L_{ij} - L_{ij}^0\right)^2 \ .$$

Ici la dénomination L^0 désigne la longueur dans la position de référence après déformation de l'utilisateur, et k_{ij} représente la raideur du ressort entre les sommets i et j. Généralement la seconde somme se réduit à 4 ou 8 voisins uniquement pour chaque sommet i, les autres k_{ij} étant nulles.

L'algorithme de minimisation sous contrainte mis en oeuvre peut alors être exprimé par

$$\begin{cases} \min E_{\text{spring}}((\mathbf{x}_i)_{i\in[\![1,N_s]\!]}) \\ \\ \text{subject to } \left|V((\mathbf{x}_i)_{i\in[\![1,N_s]\!]}) - V^0\right| = 0 \end{cases}$$

où V^0 représente le volume déformé par l'utilisateur.

Les auteurs utilisent également la méthode des multiplicateurs de Lagrange en exprimant le Lagrangien par

$$L = E_{\text{spring}} - \lambda \left|V - V^0\right| \ ,$$

où l'on souhaite obtenir $L_{,\lambda} = \left|V - V^0\right| = 0$ et $L_{,\mathbf{x}} = E_{,\mathbf{x}} - \lambda V_{,\mathbf{x}} = 0$.

L'équation du volume étant connue analytiquement en fonction des positions des sommets de la surface, les fonctions sont envoyées dans un optimiseur afin de modifier la position des points de contrôle. La convergence de la méthode peut cependant être lente compte tenu du grand nombre de sommets à prendre en compte. Pour cela, les auteurs utilisent une approche multi-résolution permettant d'initialiser l'optimisation sur des formes à basse résolution. Ensuite, les formes à résolution plus fine sont déjà proches des formes optimales ce qui permet un meilleur conditionnement des matrices. Enfin, on notera que pour améliorer la convergence, les auteurs utilisent un terme de pénalité supplémentaire dépendant du carré de la variation de volume.

Le temps de convergence obtenu est alors très faible ($\leq 200ms$) sur un processeur de $200Mhz$ de l'époque ce qui permet une déformation interactive pour de larges déformations. On peut cependant remarquer que l'optimisation ayant lieu après le placement des points de contrôle par l'utilisateur, la forme optimisée peut alors être assez différente de celle attendue si la variation de volume est importante.

3.2.3 Cas des surfaces multirésolution

Dans le cadre de déformations de surfaces multirésolution, Sauvage *et al.* [SHB07, SHBE06, Sau05] définissent une méthode de préservation du volume. Pour cela, ils définissent le volume de la surface par l'éq. 3.4 et définissent une minimisation en utilisant la méthode des multiplicateurs de Lagrange.

La méthode est notamment développée pour le cas des surfaces B-splines et surfaces de subdivision. Ensuite, celle-ci est approfondie afin de pouvoir appliquer cette contrainte de volume sur l'ensemble des échelles de la représentation multirésolution formé par des patchs B-Splines ou un schéma de subdivision.

L'intérêt de cette approche est qu'elle utilise la trilinéarité de l'expression du volume. En décomposant la correction à appliquer en trois étapes successives sur les trois axes (x, y, z), cela permet d'obtenir une expression linéaire. Une solution analytique est donc obtenue ce qui permet son implémentation très rapide.

Enfin, on notera également que cette approche est également introduite pour des surfaces maillées quelconques, où cette fois, la contrainte est intégrée après avoir linéarisé les termes au premier ordre.

Cette méthode possède un intérêt de par son application en temps réel. Pour cela, elle sera détaillée par la suite dans la partie 4.3. Et constituera le point de départ de notre méthode de correction.

Déformation de l'espace

Enfin, plus récemment, des méthodes de déformations générales applicables à des surfaces maillées quelconques commencent à se développer. De nombreuses méthodes se basent sur les déformations de l'espace.

Une méthode de déformation intéressante de par son fondement mathématique basé sur les opérations dans les espaces de Lie [Mar98] est introduite par Angelidis *et al.* [AWC04]. La déformation se réalise alors par la méthode d' "exponential mapping" permettant de gérer des paramètres de transformation robustes et cohérents avec la déformation attendue visuellement [Ale02]. Cette méthode a pour but de modeler une forme en partant de rien plutôt que de déformer un maillage déja existant (ex. argile ou pâte à modeler). Les auteurs mettent en place une méthode interactive qu'ils dénomment "Sweepers" pour la déformation basée sur le geste temps-réel d'un utilisateur. La position d'un curseur suivant ce geste est alors associée à une fonction de distance qui pondère la déformation de l'espace modelant l'objet. En notant M la matrice de transformation (translation, agrandissement ou rotation), ϕ le champ de pondération de la déformation dépendant de la distance du sommet au curseur, t_i l'instant courant, la déformation de t_i à $t_i + \Delta t$ est donnée par

$$\mathbf{x}(t_i + \Delta t) = (\phi \odot \mathrm{M})\ \mathbf{x}(t_i)\ ,$$

où l'on note $\phi \odot \mathrm{M}$ l'opération matricielle $e^{\phi \log \mathrm{M}}$. Dans un premier temps, les auteurs définissent une façon d'éviter les auto-intersections lors de la déformation en utilisant suffisamment d'itérations. Enfin, dans un second temps [ACWK04], la méthode est approfondie afin de réaliser une déformation conservant le volume. Pour cela, ils utilisent la déformation appleée "swirl" qui courbe l'espace localement sans modification de volume. Ils démontrent que le déterminant du Jacobien de la transformation est unitaire ce qui prouve que la conservation de volume est bien réalisée localement. Ainsi, une déformation modelée avec suffisamment d'itérations permet de contraindre le volume avec une très bonne approximation même dans le cas de déformations très larges (moins de 5% de déformation pour une utilisation temps réel).

Dans une autre approche de déformation de l'espace non plus dans l'optique de modeler une forme, mais dans celui de déformer un maillage, Von Funck *et al.* [FTS06] définissent une région de déformation locale autour d'un curseur que l'on peut contrôler de façon interactive. Cette région sphérique séparée en deux parties est définie par un champ vectoriel. Dans sa première partie, le champ permet de déformer l'objet dans la direction choisie du curseur (typiquement un champ quasi uniforme pour une translation) alors que dans la seconde partie sphérique, les lignes de champ se referment toutes de façon à obtenir un champ dont la divergence est nulle. Pour construire ce champ $\mathbf{v}$, les auteurs utilisent une définition par produit vectoriel[1] $\mathbf{v} = \nabla p \times \nabla q$,

[1] Permettant d'assurer une divergence nulle

où p et q sont des champs scalaires C^2. Les auteurs préconisent l'emploi des fonctions

$$p(\mathbf{x}) = \begin{cases} e(\mathbf{x}) & si \quad \|\mathbf{x}-\mathbf{x}_0\| \in [0, r_0] \\ (1-b)\, e(\mathbf{x}) & si \quad \|\mathbf{x}-\mathbf{x}_0\| \in [r_0, r_1] \\ 0 & sinon \end{cases}$$

et

$$q(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & si \quad \|\mathbf{x}-\mathbf{x}_0\| \in [0, r_0] \\ (1-b)\, f(\mathbf{x}) & si \quad \|\mathbf{x}-\mathbf{x}_0\| \in [r_0, r_1] \\ 0 & sinon \end{cases}$$

avec $\mathbf{x}_0$ la position centrale du curseur, r_0 le rayon intérieur de la région sphérique et r_1 le rayon extérieur au delà duquel le champ sera nul. La fonction de poids b est choisie lisse et variant entre 1 et 0. Les auteurs proposent notamment l'utilisation de polynômes de Bernstein du troisième degré. Enfin, les fonctions e et f sont définies suivant la déformation voulue dans la région centrale.

Dans le cas d'advection par un champ homogène $\mathbf{v}(\mathbf{x}) = \mathbf{v}_0$ tel que $\|\mathbf{v}_0\| = 1$, on choisit deux vecteurs normés $\mathbf{v}_1^T$ et $\mathbf{v}_2^T$ perpendiculaires à $\mathbf{v}_0$ et on construit

$$\begin{cases} e(\mathbf{x}) = \mathbf{v}_1^T \cdot (\mathbf{x}-\mathbf{x}_0) \\ f(\mathbf{x}) = \mathbf{v}_2^T \cdot (\mathbf{x}-\mathbf{x}_0) \end{cases}$$

Dans le cas d'une rotation à l'intérieur de la région (à rapprocher du "swirl" de [ACWK04]) autour du point $\mathbf{x}_0$ selon l'axe $\mathbf{a}$, les auteurs suggèrent de définir

$$\begin{cases} e(\mathbf{x}) = \mathbf{a} \cdot (\mathbf{x}-\mathbf{x}_0) \\ f(\mathbf{x}) = \|\mathbf{a} \times (\mathbf{x}-\mathbf{x}_0)\|^2 \end{cases}$$

La déformation ayant lieu le long du champ vectoriel ainsi défini, il est nécessaire d'intégrer numériquement l'équation différentielle ordinaire le long des lignes de champ. Le très classique Runge-Kutta d'ordre quatre est utilisé dans ce cas. Le cas de larges déformations est bien pris en compte, mais nécessite une méthode de remaillage. L'avantage principal de la méthode est que, de par la définition même de l'outil de modélisation par ce type de champ vectoriel, les autocollisions sont évitées. De plus le champ étant de divergence nulle, le volume est conservé de façon locale automatiquement. Leurs résultats montrent une bonne précision numérique de la conservation du volume avec la méthode d'intégration choisie. De plus les calculs sont suffisamment rapides pour obtenir une déformation en temps réel avec les cartes graphiques actuelles.

Le contrôle de la déformation n'est cependant pas forcément adapté au cas d'animation de personnages. Si celui-ci permet de déformer un maillage dans une pose donnée avec succès, il parait fastidieux, dans le cas d'animations, de définir pour chaque partie du personnage à déplacer un champ vectoriel dont l'influence et le type sont à redéfinir. En effet, il est nécessaire de définir les zones d'influences de la déformation de façon volumique (sphère, cylindre, ...) pour chaque partie à déformer.

Une façon d'améliorer la prise en main de ces outils, serait par exemple de lier ces zones à une distance topologique le long de la surface ce qui rendrait ces sélections plus automatiques.

Utilisation du système de coordonnées différentielles

Récemment, une méthode de codage de surface s'est avérée particulièrement efficace pour effectuer des déformations. Pour cela, la surface n'est plus définie par ses coordonnées cartésiennes, mais par une représentation intrinsèque ne dépendant pas de sa position dans l'espace. Un exemple de ce codage s'est développé sous la dénomination de coordonnées laplaciennes [CS06] et décrit la surface de façon différentielle. Un résumé de la méthode est notamment donné par

Sorkine [Sor06]. On note alors les coordonnées laplaciennes d'un sommet s possédant un voisinage $\mathcal{N}$ par $\boldsymbol{\delta}_s$ tel que

$$\boldsymbol{\delta}_s = \mathbf{x}_s - \frac{1}{\#\mathcal{N}} \sum_{\mathbf{x}_{s'} \in \mathcal{N}} \mathbf{x}_{s'} \ ,$$

où $\#\mathcal{N}$ représente le degré de valence du sommet s. Cette représentation permet d'approximer de façon discrète l'opérateur laplacien sur la surface[2] et donne une information sur la courbure locale de la surface [MDSB02]. Cette décomposition peut s'exprimer sous forme matricielle. Ainsi la reconstruction de la surface se réalise en résolvant un système linéaire pour lequel il faut rajouter l'information de translation en coordonnées cartésiennes.

Sorkine *et al.* [SCL+04] ainsi que Lipman *et al.* [LSC+04] proposent alors de déformer la surface dans cet espace de coordonnées qui a pour avantage de préserver les détails de la forme. On déforme dans une première partie le maillage en fixant manuellement la position d'un nombre N_1 de sommets $(\mathbf{x}_{s_i})_{i \in [\![1,N_1]\!]}$. Dans un second temps, on déforme l'ensemble du maillage en minimisant l'écart entre les coordonnées laplaciennes originales et les nouvelles tout en tentant de rester au plus proche des points nouvellement déplacés par l'utilisateur (au sens de la norme euclidienne). On note $\mathbf{x}^0$ les positions euclidiennes initiales du maillage et $\mathbf{x}^1$ les positions après déformation manuelle. On suppose que le maillage contient N_0 sommets et ceux qui sont modifiés sont désignés par l'ensemble d'indices $\mathcal{I}$. Une fonctionnelle d'énergie est définie par

$$\begin{aligned} E((\mathbf{x}_s)_{s \in [\![1,N_0]\!]}) = & \sum_{s=1}^{N_0} \left\| \delta(\mathbf{x}_s^0) - \delta(\mathbf{x}_s) \right\|^2 \\ & + \sum_{s \in \mathcal{I}} \|\mathbf{x}_s - \mathbf{x}_s^1\|^2 \ . \end{aligned}$$

Les auteurs mentionnent cependant que la résolution de cette minimisation ne permet pas de réaliser des rotations et agrandissements des détails de la structure du fait de la limitation des coordonnées laplaciennes. En effet celles-ci ne sont pas invariantes par rapport à la rotation. La minimisation dans l'espace des coordonnées laplaciennes ne se réalise alors plus uniquement avec la comparaison des coordonnées initiales $\delta(\mathbf{x}_s^0)$ mais avec une transformation affine T_s de ces coordonnées effectuée sur chaque sommet. La première somme de la minimisation est alors remplacée par

$$\sum_{s=1}^{N_0} \left\| T_s(\mathbf{x}) \delta(\mathbf{x}_s^0) - \delta(\mathbf{x}_s) \right\|^2 \ .$$

Ici cependant, T_s est une inconnue supplémentaire dans l'équation. Pour cela, les auteurs considèrent que T doit pouvoir contenir une homothétie et une rotation et définissent donc une forme spéciale de cette matrice. Ensuite, ils suggèrent que pour chaque sommet s, cette matrice soit définie de sorte à minimiser

$$\|T_s \, \mathbf{x}_s^0 - \mathbf{x}_s\|^2 + \sum_{j \in \mathcal{N}} \|T_s \mathbf{x}_j^0 - \mathbf{x}_j\|^2 \ .$$

Cette méthode, permet de déformer un maillage préexistant et les auteurs donnent des exemples dans le cas de déformations importantes de maillages complexes. Ils ont également mis en place un outil de déformation interactif basé sur des courbes [NSACO05] représentant le profil que la surface doit prendre. Cependant, la localité de la déformation ne peut pas être contrôlée manuellement par l'artiste.

On pourra remarquer que la limitation sur la non invariance par rotation des coordonnées laplaciennes peut être surmontée par l'introduction d'un autre système de coordonnées différentielles discret défini rigoureusement dans [LSLC05]. Ce système permet alors de définir la surface indépendamment de sa position et de son orientation, et permet alors d'y réaliser des déformations

[2] On parle également dans le cas général d'opérateur de Laplace-Beltrami sur une variétée Riemannienne.

sans devoir utiliser la précaution supplémentaire de la transformation affine intermédiaire. Dans une autre approche, Yu *et al.* [YZX+04] ainsi que Xu *et al.* [XZWB05] utilisent une déformation basée sur l'équation de Poisson qui possède également des propriétés différentielles [XZB06]. La déformation est également invariante par rotation, bien que les nouvelles méthodes laplaciennes semblent donner des résultats plus probants.

De plus, il est intéressant de remarquer que Huang *et al.* [HSL+06] réutilisent la méthode des coordonnées laplaciennes et ajoutent une contrainte de rigidité compatible avec les modèles à squelettes que nous souhaitons utiliser. Ainsi lors de la déformation articulée par un squelette, les différentes parties doivent rester relativement droites. Le squelette n'est alors pas directement utilisé pour l'animation mais est employé en tant que contrainte de façon à préserver l'aspect rigide ainsi que la longueur des sections. L'ensemble de la déformation est alors traité en résolvant un grand système linéaire creux. Récemment, Masuda *et al.* ont présenté une méthode de préservation de caractéristiques de la forme en contraignant le système à résoudre [MYF06] afin de pouvoir utiliser les déformations dans le cas de modèles rigides (carrosseries de voitures, ...).

Enfin, plusieurs approches ont été effectuées afin de contraindre ce type de déformation à garder un volume constant. Zhou *et al.* [ZHS+05] définissent ainsi une méthode leur permettant de limiter les variations de volume. Pour cela, ils considèrent une approche volumique de la forme en construisant un maillage intérieur à la surface. L'étape de maillage est définie de façon simple de sorte à limiter le temps de précalcul. Ensuite, la minimisation de la déformation est calculée en prenant en compte une contrainte sur la déformation du maillage intérieur (volumique). Cela permet alors d'obtenir des résultats qui visuellement conservent mieux le volume que la prise en compte unique de la surface. On peut cependant noter que le volume n'est par contre pas respecté de façon exacte, la méthode ne permettant que d'améliorer l'aspect visuel.

De façon plus exacte, Huang *et al.* [HSL+06] prennent en compte le volume défini par la surface en utilisant l'éq. 3.4. Ils ajoutent alors cette contrainte directement dans l'équation de minimisation de leur déformation. La contrainte étant non linéaire, les auteurs définissent la linéarisation ainsi que la méthode itérative de résolution. Cette fois, la déformation prend en compte la contrainte de volume de façon exacte bien que la méthode de résolution soit assez complexe à mettre en place dans le cadre d'une utilisation interactive.

Enfin Lipman *et al.* [LCOGL06] proposent une autre approche permettant des déformations importantes avec préservation de volume. L'approche est plus originale car elle consiste à paramétrer les rotations de l'espace dans le cas d'utilisation des coordonnées laplaciennes. Les auteurs définissent alors l'expression du volume local suivant les variations de la normale à la surface. L'approche possède l'avantage de contrôler le volume de façon locale (et donc globale également). Même pour des déformations très importantes, le volume est préservé de façon correcte par la minimisation. Un large système creux doit cependant toujours être résolu à chaque déformation.

Déformer la surface dans un système de coordonnées différentielles permet donc d'obtenir des résultats convaincants de par la prise en compte directe de l'interaction des voisins. Des extensions permettent également d'ajouter différentes contraintes (rigidités, volumes, ...). Les méthodes de déformations peuvent enfin désormais être mises en place de façon interactive grâce à des méthodes itératives rapides d'inversion matricielle. Cependant, si elles permettent de déformer une surface d'une pose à l'autre de façon locale, la méthode de skinning de par son organisation hiérarchique du squelette reste la plus efficace pour mettre en place rapidement et intuitivement le mouvement animé d'un personnage sans avoir à positionner séparément chaque pose.

Chapitre 4

Application de la contrainte de volume au skinning

Nous allons maintenant présenter les résultats obtenus pour l'application de la contrainte de volume sur la surface animée par squelette. Dans un premier temps, nous observerons l'effet du choix des poids de skinning sur l'aspect d'un modèle de cylindre défini paramétriquement, La méthode de base de contrainte de volume sur les surfaces triangulées sera par la suite introduite sous la forme d'une minimisation sous contrainte. Une première résolution sera réalisée dans un cas très général. Par la suite, chacune des différentes parties tâchera d'améliorer un ou plusieurs points des défauts constatés.

4.1 Exemple du choix des poids de skinning

Cette première partie ne s'intéresse pour l'instant qu'à l'aspect de la déformation pour différents choix de calcul des poids de skinning. A l'opposé des sections suivantes, l'étude se réalise directement sur une surface paramétrique et non sur un maillage triangulé. Cette partie ne constitue qu'un point de départ à une étude plus poussée tentant d'établir une méthode de correction par une approche analytique.

4.1.1 Modèle du cylindre

Dans un premier temps, nous allons tenter de comprendre l'effet du soft skinning classique sur un cas algébrique pour une surface continue. On considère les deux cylindres avec deux joints dont l'un effectue une rotation pouvant correspondre au cas du pliement du coude. Nous définissons cette fois la surface paramétriquement et essayons de calculer de façon algébrique la variation du volume de celle-ci en fonction de l'angle de la rotation.

La Fig. 4.1 montre le modèle initial utilisé pour les deux cylindres.

4.1.2 Équation de la surface skinnée

La surface globale est formée par les deux sous-surfaces $\mathbf{s}_1$ et $\mathbf{s}_2$ telles que

$$\mathbf{s}_1 : (u_1, u_2) \rightarrow \begin{pmatrix} R\cos(u_1) \\ R\sin(u_1) \\ u_2 \end{pmatrix} \text{ et}$$

$$\mathbf{s}_2 : (u_1, u_2) \rightarrow \begin{pmatrix} R\cos(u_1) \\ R\sin(u_1) \\ u_2 + L \end{pmatrix}$$

pour $u_1 \in [0, 2\pi]$, et $u_2 \in [0, L]$.

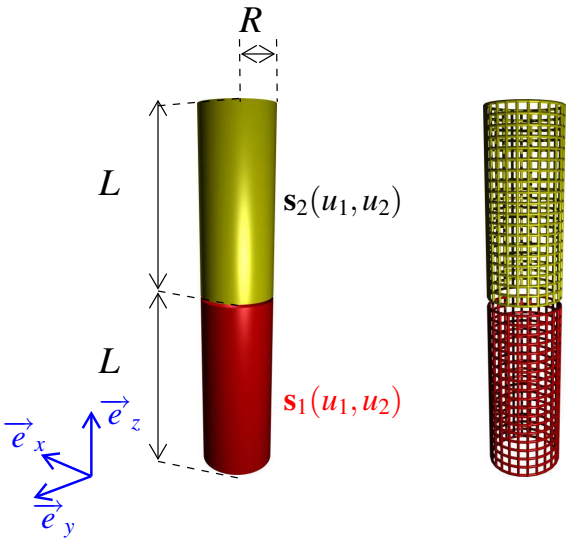


Fig. 4.1: Modèle des deux cylindres utilisés. La figure de gauche représente la surface alors que celle de droite montre la paramétrisation utilisée.

La distance d'un point de s_1 par rapport au segment du squelette lié au premier repère est de $d_1^{s_1} = R$, alors que la distance au segment lié repère deux du squelette est donnée par $d_2^{s_1} = \sqrt{R^2 + (L - u_2)^2}$.

De même, la distance d'un point de s_2 par rapport au premier et second repère est donné par $d_1^{s_2} = \sqrt{R^2 + u_2^2}$ et $d_2^{s_2} = R$.

On considère des poids de skinning proportionnels à l'inverse de ces distances. On note alors

$$\omega_j^{s_i} = \frac{(d_j^{s_i})^{-\alpha}}{(d_1^{s_i})^{-\alpha} + (d_2^{s_i})^{-\alpha}}$$

On obtient donc

$$\begin{cases} \omega_1^{s_1} = \dfrac{\sqrt{R^2 + (L - u_2)^2}^{\alpha}}{R^\alpha + \sqrt{R^2 + (L - u_2)^2}^{\alpha}} \\ \\ \omega_2^{s_1} = \dfrac{R^\alpha}{R^\alpha + \sqrt{R^2 + (L - u_2)^2}^{\alpha}} \end{cases}$$

et

$$\begin{cases} \omega_1^{s_2} = \dfrac{R^\alpha}{R^\alpha + \sqrt{R^2 + u_2^2}^{\alpha}} \\ \\ \omega_2^{s_2} = \dfrac{\sqrt{R^2 + u_2^2}^{\alpha}}{R^\alpha + \sqrt{R^2 + u_2^2}^{\alpha}} \end{cases}$$

On note ensuite les paramètres de changement de repère. On note t l'opérateur de changement de base du repère local du second joint vers le repère global (ici le premier joint). On a donc dans le cas d'une rotation d'angle θ autour de l'axe y

$$\begin{cases} t & : (x, y, z) \mapsto (x, y, z + L) \\ t^{-1} & : (x, y, z) \mapsto (x, y, z - L) \end{cases}$$

On définit également la transformation du second joint exprimé dans son repère local. On a donc

$$\mathcal{R}_\theta : (x, y, z) \mapsto \begin{pmatrix} x\cos(\theta) + z\sin(\theta) \\ y \\ -x\sin(\theta) + z\cos(\theta) \end{pmatrix}$$

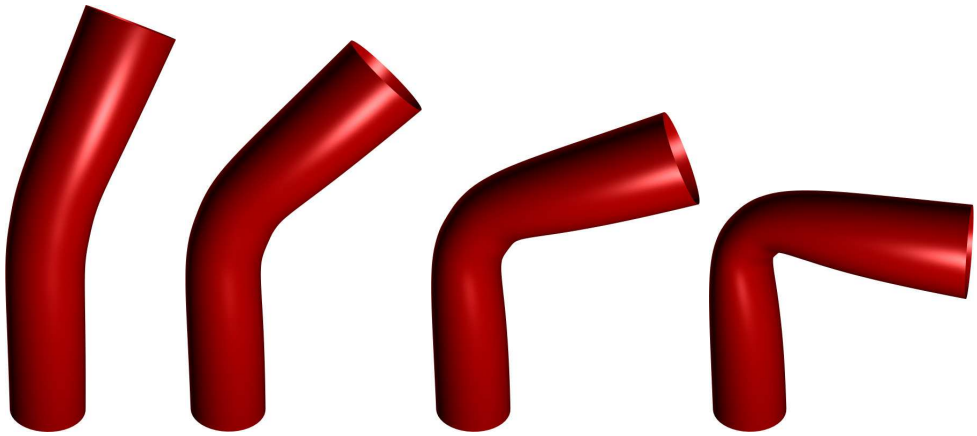


Fig. 4.2: Cylindres déformés pour différents angles pour $\alpha = 2$. De gauche à droite, l'angle de déformation est de $\pi/8, \pi/4, 3\pi/8$ et $\pi/2$.

La surface obtenue sans déformation par skinning lisse est donc obtenue simplement par les deux sous-surfaces $\mathbf{S}_1$ et $\mathbf{S}_2$ avec

$$\left\{ \begin{array}{l} \mathbf{S}_1(u_1, u_2) = \mathbf{s}_1(u_1, u_2) \\ \mathbf{S}_2(u_1, u_2) = t \circ \mathcal{R}_\theta \circ t^{-1} \circ \mathbf{s}_2(u_1, u_2) \end{array} \right. \tag{4.1}$$

pour $(u_1, u_2) \in [0, 2\pi] \times [0, L]$.

Alors que dans le cas du skinning lisse, on a maintenant la combinaison

$$\left\{ \begin{array}{l} \mathbf{S}_1 = \left(\omega_1^{s_1}\ \mathrm{I} + \omega_2^{s_1}\ t \circ \mathcal{R}_\theta \circ t^{-1}\right) \circ s_1 \\ \mathbf{S}_2 = \left(\omega_1^{s_2}\ \mathrm{I} + \omega_2^{s_2}\ t \circ \mathcal{R}_\theta \circ t^{-1}\right) \circ s_2 \end{array} \right. \tag{4.2}$$

avec I la fonction identité.

On peut donc écrire l'équation de la surface de façon analytique.

$$\begin{aligned} \mathbf{S}_1 = & \frac{\sqrt{R^2+(L-u_2)^2}^\alpha}{R^\alpha+\sqrt{R^2+(L-u_2)^2}^\alpha} \begin{pmatrix} R\ \cos(u_1) \\ R\ \sin(u_1) \\ u_2 \end{pmatrix} \\ + & \frac{R^\alpha}{R^\alpha+\sqrt{R^2+(L-u_2)^2}^\alpha} \\ & \begin{pmatrix} R\ \cos(u_1)\ \cos(\theta) + (u_2 - L)\ \sin(\theta) \\ R\ \sin(u_1) \\ -R\ \cos(u_1)\ \sin(\theta) + (u_2 - L)\ \cos(\theta) + L \end{pmatrix} \\ \mathbf{S}_2 = & \frac{R^\alpha}{R^\alpha+\sqrt{R^2+u_2^2}^\alpha} \begin{pmatrix} R\ \cos(u_1) \\ R\ \sin(u_1) \\ u_2 + L \end{pmatrix} \\ + & \frac{\sqrt{R^2+u_2^2}^\alpha}{R^\alpha+\sqrt{R^2+u_2^2}^\alpha} \\ & \begin{pmatrix} R\ \cos(u_1)\ \cos(\theta) + u_2\ \sin(\theta) \\ R\ \sin(u_1) \\ -R\ \cos(u_1)\ \sin(\theta) + u_2\ \cos(\theta) + L \end{pmatrix} \end{aligned}$$

Enfin, on notera que la surface se referme par les deux disques aux extrémités $\mathbf{d}_1$ et $\mathbf{d}_2$. Pour ces disques non visibles normalement de l'extérieur, on ne va pas considérer de skinning spécial

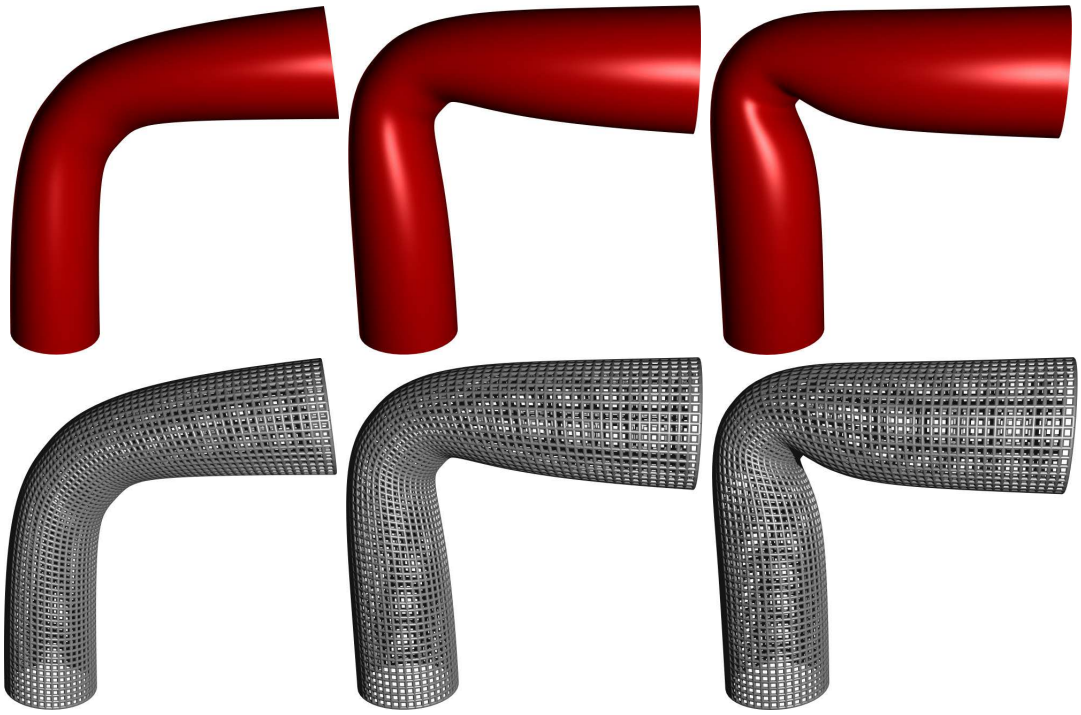


Fig. 4.3: Action du paramètre α sur la localité de la déformation du cylindre. Les trois déformations correspondent respectivement à α =1, 2 et 4. Les figures du dessus montrent la surface alors que celles du dessous montrent l'effet sur un maillage.

à l'intérieur du disque en fonction de la distance au squelette. Seuls les poids déjà utilisés avec $u_2 = 0$ ou L sont pris en compte. Les expressions s'obtiennent alors similairement avec

$$\begin{aligned}
\mathbf{d}_1(u_1', u_2') &= \frac{\sqrt{R^2+L^2}^{\alpha}}{R^\alpha+\sqrt{R^2+L^2}^{\alpha}} \begin{pmatrix} u_2' \cos(u_1') \\ u_2' \sin(u_1') \\ 0 \end{pmatrix} \\
&+ \frac{R^\alpha}{R^\alpha+\sqrt{R^2+L^2}^{\alpha}} \\
&\begin{pmatrix} u_2' \cos(u_1')\cos(\theta) - L\sin(\theta) \\ u_2' \sin(u_1') \\ -u_2' \cos(u_1')\sin(\theta) - L\cos(\theta) + L \end{pmatrix} \\
\mathbf{d}_2(u_1', u_2') &= \frac{R^\alpha}{R^\alpha+\sqrt{R^2+L^2}^{\alpha}} \begin{pmatrix} u_2' \cos(u_1') \\ u_2' \sin(u_1') \\ 2\,L \end{pmatrix} \\
&+ \frac{\sqrt{R^2+L^2}^{\alpha}}{R^\alpha+\sqrt{R^2+L^2}^{\alpha}} \\
&\begin{pmatrix} u_2' \cos(u_1')\cos(\theta) + L\sin(\theta) \\ u_2' \sin(u_1') \\ -u_2' \cos(u_1')\sin(\theta) + L\cos(\theta) + L \end{pmatrix}
\end{aligned}$$

avec $(u_1', u_2') \in [0, 2\pi] \times [0, L]$.

On illustre en Fig. 4.2 la déformation du cylindre pour différents angles.

On peut également observer l'influence du paramètre α. Plus celui-ci est élevé, plus la déformation va être locale, mais également plus visible. La Fig. 4.3 montre l'évolution de la déformation selon le paramètre α.

4.1.3 Calcul du volume

Cas des cylindres

On peut maintenant exprimer le volume englobé de façon exacte. La normale de chaque surface $\mathbf{s}(u_1, u_2)$ est donnée par

$$\mathbf{n_s}(u_1, u_2) = \frac{\partial \mathbf{s}}{\partial u_1}(u_1, u_2) \times \frac{\partial \mathbf{s}}{\partial u_2}(u_1, u_2)$$

La contribution des deux disques $\mathbf{d}_1$ et $\mathbf{d}_2$ se calcule aisément pour obtenir

$$\begin{cases} \mathbf{n_{d_1}} \cdot \mathbf{e}_3 = & - \left(\omega_1^{\mathbf{d}_1} + \omega_2^{\mathbf{d}_1} \cos(\theta)\right) u_2 \\ \mathbf{n_{d_2}} \cdot \mathbf{e}_3 = & \left(\omega_1^{\mathbf{d}_2} + \omega_2^{\mathbf{d}_2} \cos(\theta)\right) u_2 \end{cases}$$

en ayant noté l'identité $\omega_1 + \omega_2 = 1$.

La contribution de volume de ces deux disques est alors donnée par

$$\begin{aligned} V_{\mathbf{d}_1} = & -\int_{u_1=0}^{2\pi} \int_{u_2=0}^{R} \omega_2^{\mathbf{d}_1} \Big[(1 - \cos(\theta))\ L - \\ & u_2 \cos(u_1) \sin(\theta) \Big] \left(\omega_1^{\mathbf{d}_1} + \omega_2^{\mathbf{d}_1} \cos(\theta)\right) \\ & u_2\ \mathrm{d}u_1 \mathrm{d}u_2\ . \end{aligned}$$

Ce qui revient, tout calcul réalisé à

$$V_{\mathbf{d}_1} = -\pi\, R^2\, L\, \omega_2^{\mathbf{d}_1} \left(\omega_1^{\mathbf{d}_1} + \omega_2^{\mathbf{d}_1} \cos(\theta)\right) (1 - \cos(\theta))\ .$$

De la même façon, nous pouvons obtenir pour $\mathbf{d}_2$

$$\begin{aligned} V_{\mathbf{d}_2} = & \ \pi\, R^2\, L \left(\omega_1^{\mathbf{d}_2} + \omega_2^{\mathbf{d}_2} \cos(\theta)\right) \\ & \left(2\, \omega_1^{\mathbf{d}_2} + \omega_2^{\mathbf{d}_2}\ (1 + \cos(\theta))\right)\ . \end{aligned}$$

On peut bien vérifier que pour un angle $\theta = 0$, la contribution du premier disque au volume est nulle (disque confondu avec le plan $z = 0$), alors que celle du second est de $V_{\mathbf{d}_2}(\theta = 0) = 2\pi R^2 L$ qui correspond bien au volume du cylindre de hauteur $2\,L$.

Les deux contributions des volumes englobés par les deux disques s'ajoutent ce qui fournit dans ce cas

$$\begin{aligned} & V_{\mathbf{d}_1}(\theta) + V_{\mathbf{d}_2}(\theta) = \\ & \pi R^2\, L \frac{1}{\left(R^\alpha + \sqrt{R^2 + L^2}^\alpha\right)^2} \\ & \Big[\left(R^\alpha + \sqrt{R^2 + L^2}^\alpha \cos(\theta)\right) \\ & \left(2\, R^\alpha + \sqrt{R^2 + L^2}^\alpha\, (1 + \cos(\theta))\right) \\ & - \\ & R^\alpha \left(R^\alpha \cos(\theta) + \sqrt{R^2 + L^2}^\alpha\right) (1 - \cos(\theta)) \Big] \end{aligned}$$

Le cas des deux cylindres $\mathbf{S}_1$ et $\mathbf{S}_2$ est plus complexe, et l'intégration ne s'exprime pas analytiquement. Une méthode d'intégration numérique est alors nécessaire.

4.1.4 Poursuite des travaux

Ce travail nécessite d'être poursuivi dans le cadre de la correction de volume. Une fois la perte de volume exprimée en fonction de l'angle de la courbure du cylindre, il pourra être intéressant de définir une méthode de déformation de la surface par "gonflement" local de la surface au niveau du coude.

Il pourrait également être intéressant de définir les poids de skinning non plus par rapport à une distance volumique mais par une distance définie de façon topologique (distance géodésique sur la surface).

4.2 Méthode générale de correction de volume

La contrainte de volume appliquée sur une surface triangulée va maintenant être introduite. Pour cela, nous choisissons de réaliser la correction en deux étapes à chaque instant t.

1. Une étape de déformation par skinning classique

$$\forall s \in [\![1, N_s]\!], \quad \mathbf{x}_s(t) = \left(\sum_i \omega_s^i \, T^i(t) \right) \underline{\mathbf{x}}_s^0 \ .$$

2. Une étape de correction par minimisation sous contrainte.

   D'une façon générale, on recherchera une correction donnée par le vecteur $\mathbf{u} = (\mathbf{u}^x, \mathbf{u}^y, \mathbf{u}^z) \in \mathbb{R}^{3\,N_s}$. Ce vecteur de correction devra être de faible norme, et permettre de corriger le volume.

   On appelle pour simplifier $V = V(\mathbf{x})$ le volume englobé par la surface obtenue après application du skinning. Et $V^0 = V(\underline{\mathbf{x}}^0)$, le volume initial de la forme que l'on souhaiterait conserver. On appelle également pour une pose donnée $\Delta V = V^0 - V(\mathbf{x})$, l'écart de volume entre la surface skinnée, et originale.

   La minimisation pourra donc s'écrire sous la forme

$$\begin{cases} \min & \|\mathbf{u}\|^2 \\ \text{contraint à} & V(\mathbf{x} + \mathbf{u}) = V^0 \ . \end{cases} \tag{4.3}$$

   On notera cependant que la fonction à minimiser (ici $f : \mathbf{u} \mapsto \|\mathbf{u}\|^2$) peut être modifiée. En effet, c'est ce choix de minimisation qui va dicter l'aspect et le comportement de la surface obtenue après la résolution de la minimisation. Ainsi plusieurs autres types de fonctions (positives et convexes) peuvent permettre de contrôler l'aspect et les caractéristiques de la solution obtenue (ex. contrôle de l'aspect lisse de la surface en prenant en compte ses dérivées, pondération, ...). La contrainte stricte de volume constant est quant à elle fixe, et on ne cherchera pas à modifier cette condition.

   Afin de résoudre cette expression, nous allons utiliser la méthode des multiplicateurs de Lagrange. On introduit dans ce cas le paramètre supplémentaire λ et l'on réecrit le problème sous la forme d'une fonctionnelle d'énergie

$$E(\mathbf{u}, \lambda) = \|\mathbf{u}\|^2 + \lambda \left[V(\mathbf{x} + \mathbf{u}) - V^0 \right] \ . \tag{4.4}$$

   Le problème de minimisation sous contrainte initial est alors équivalent à un problème de point selle (min max) où la solution optimale est donnée par

$$\min_{\mathbf{u}} \max_{\lambda \in \mathbb{R}} E(\mathbf{u}, \lambda) \ .$$

   Si la solution de ce problème existe, alors il est nécessaire que celle-ci annule le gradient de E (extremum de E).

$$\nabla E(\mathbf{u}, \lambda) = 0 \ . \tag{4.5}$$

Ainsi, la résolution des problèmes de minimisation posés dans les sections suivantes sera ramenée à la résolution du système donné par l'annulation des dérivées partielles par rapport aux composantes de l'inconnue $\mathbf{u}$ et au paramètre de contrainte stricte λ.

Nous choisissons de résoudre la contrainte par cette méthode car elle possède l'avantage de pouvoir exprimer la solution au problème de façon algébrique ce qui est un avantage dans le cas d'une implémentation rapide de la résolution.

4.3 Cas linéarisé

Linéarisation

Nous allons tout d'abord nous intéresser au cas où la contrainte de volume va être linéarisée au premier ordre. Cette méthode est directement inspirée de [Sau05] pour le cas de surfaces triangulées.

On rappelle que le volume non corrigé obtenu par skinning est donné par (voir eq. 3.5)

$$V(\mathbf{x},\mathbf{y},\mathbf{z}) = \sum_{(i,j,k)} \beta_{ijk}\, x_i y_j z_k \ .$$

Après correction par le vecteur $\mathbf{u}$, le volume est alors donné par $V(\mathbf{x}+\mathbf{u}^x, \mathbf{y}+\mathbf{u}^y, \mathbf{z}+\mathbf{u}^z)$.

On va maintenant supposer que la correction est d'amplitude faible de sorte à pouvoir négliger les termes quadratiques et le terme cubique de l'expression du volume par rapport aux termes linéaires. Sous cette hypothèse, on peut alors décomposer le volume sous la forme

$$\begin{aligned}&V(\mathbf{x}+\mathbf{u}^x, \mathbf{y}+\mathbf{u}^y, \mathbf{z}+\mathbf{u}^z) - V(\mathbf{x},\mathbf{y},\mathbf{z}) = \\ &V(\mathbf{u}^x,\mathbf{y},\mathbf{z}) + V(\mathbf{x},\mathbf{u}^y,\mathbf{z}) + V(\mathbf{x},\mathbf{y},\mathbf{u}^z) + O(\|\mathbf{u}\|^2) \ .\end{aligned}$$

Expression du Lagrangien

On peut donc écrire au premier ordre en utilisant les notations de produits scalaires introduites dans l'eq. 3.7.

$$V(\mathbf{x}+\mathbf{u}) \simeq V(\mathbf{x}) + <\mathbf{u}^x, \mathbf{V}^{YZ}> + <\mathbf{u}^y, \mathbf{V}^{XZ}> + <\mathbf{u}^z, \mathbf{V}^{XY}> \ .$$

En introduisant désormais cette expression approchée dans l'eq. 4.4, on peut donc écrire

$$E(\mathbf{u},\lambda) = \|\mathbf{u}\|^2 + \lambda\left[<\mathbf{u}^x, \mathbf{V}^{YZ}> + <\mathbf{u}^y, \mathbf{V}^{XZ}> + <\mathbf{u}^z, \mathbf{V}^{XY}> - \Delta V\right] \ . \tag{4.6}$$

Correction du volume

Suite à la linéarisation, les corrections sur les axes ne dépendent plus les unes des autres. Il est désormais nécessaire de déterminer comment cette correction va agir sur chacun d'eux.

Dans le cas le plus général possible, on peut essayer de moyenner celle-ci équitablement sur chaque axe. Ainsi, un tiers de la correction sera porté par l'axe x, un autre tiers par l'axe y et un dernier par z. On en déduit alors que

$$\begin{cases} <\mathbf{u}^x, \mathbf{V}^{YZ}> = \dfrac{\Delta V}{3} \\[2ex] <\mathbf{u}^y, \mathbf{V}^{XZ}> = \dfrac{\Delta V}{3} \\[2ex] <\mathbf{u}^z, \mathbf{V}^{XY}> = \dfrac{\Delta V}{3} \end{cases}$$

On peut alors montrer (Annexe A) qu'en résolvant le système donné en eq. 4.5, le résultat s'exprime de façon analytique par

$$(\mathbf{u}^x, \mathbf{u}^y, \mathbf{u}^z) = \frac{\Delta V}{3} \left(\frac{\mathbf{V}^{YZ}}{\|\mathbf{V}^{YZ}\|^2}, \frac{\mathbf{V}^{XZ}}{\|\mathbf{V}^{XZ}\|^2}, \frac{\mathbf{V}^{XY}}{\|\mathbf{V}^{XY}\|^2} \right) . \tag{4.7}$$

4.4 Cas exact

La méthode précédente effectue la correction après avoir réalisé une approximation sur la contrainte de volume.

Ainsi, le volume n'est pas exactement corrigé, mais seulement approché au premier ordre. Cependant, l'erreur commise peut être connue par rapport à la variation ΔV.

Il peut toutefois être intéressant de noter que le cas de correction exacte peut également être traité. Pour cela, nous allons séparer la correction en trois minimisations à la chaîne. Le principe va consister à corriger un tiers du volume suivant un axe, puis corriger la moitié de l'erreur restante suivant un autre axe, et enfin l'erreur résiduelle suivant le dernier axe.

Évidemment, le choix de l'ordre des axes est arbitraire. Prenons le cas de l'ordre x, puis y, et enfin, suivi de z. On peut alors décomposer par exemple la méthode sous cette forme

1. L'axe x. On obtient alors $V(\mathbf{x} + \mathbf{u}^x, \mathbf{y}, \mathbf{z})$, et l'on corrige 1/3 du volume.
2. L'axe y. On obtient alors $V(\mathbf{x} + \mathbf{u}^x, \mathbf{y} + \mathbf{u}^y, \mathbf{z})$ et on corrige la moitié de l'erreur restante.
3. L'axe z. On obtient alors $V(\mathbf{x} + \mathbf{u}^x, \mathbf{y} + \mathbf{u}^y, \mathbf{z} + \mathbf{u}^z)$ et l'on corrige l'erreur résiduelle.

On peut alors montrer (Annexe B) qu'en considérant cette répartition de la correction sur les trois axes, celle-ci peut s'écrire sous la forme

$$(\mathbf{u}^x, \mathbf{u}^y, \mathbf{u}^z) = \frac{\Delta V}{3} \left(\frac{\mathbf{V}^{YZ}}{\|\mathbf{V}^{YZ}\|^2}, \frac{{}^*\mathbf{V}^{XZ}}{\|{}^*\mathbf{V}^{XZ}\|^2}, \frac{{}^{**}\mathbf{V}^{XY}}{\|{}^{**}\mathbf{V}^{XY}\|^2} \right) , \tag{4.8}$$

avec cette fois les valeurs corrigées dues à la correction de l'étape précédente sur l'autre axe telles que

$$\begin{cases} \forall j \in [\![1, N_s]\!] , \quad {}^*V_j^{XZ} = \sum_{(i,k)} \beta_{ijk} \left(x_i + u_i^x\right) z_k \\ \forall k \in [\![1, N_s]\!] , \quad {}^{**}V_k^{XY} = \sum_{(i,j)} \beta_{ijk} \left(x_i + u_i^x\right) \left(y_j + u_j^y\right) \end{cases}$$

4.4.1 Résultats

Modèle

Afin de tester les méthodes mises en place, nous prenons le cas d'exemples formés par un cylindre. Trois joints sont définis. Un en bas, un au milieu et un en haut. Le joint central subit une rotation d'angle 90° et entraîne celui du haut. Les poids sont définis automatiquement en considérant l'inverse du carré de la distance du sommet donné par rapport au premier et dernier joint. Puis ceux-ci sont normalisés afin de donner une somme unitaire sur chaque sommet. Le joint central ne sert, quant à lui, que pour le squelette et ne rentre pas en compte pour la déformation de la surface.

La déformation de skinning classique appliqué à ce modèle ainsi que la répartition des poids est montré en Fig. 4.4.

Correction de volume

On peut tout d'abord s'intéresser aux résultats numériques concernant la correction du volume.

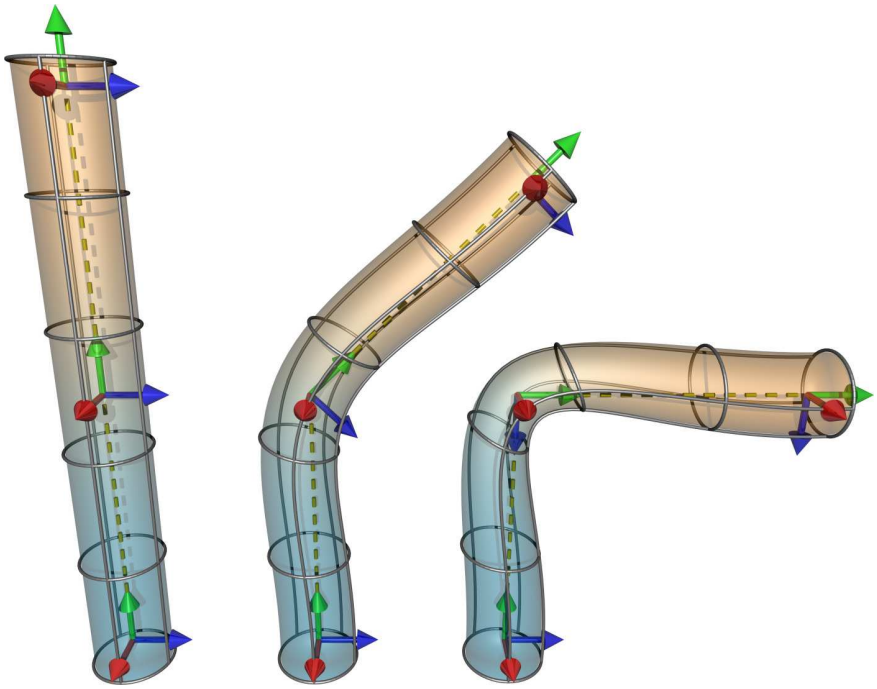


Fig. 4.4: Modèle de test du cylindre pris en compte. Le cylindre est déformé par skinning classique. Le squelette est formé de trois joints, dont seuls les deux extrêmes ont une influence sur la surface par les poids de skinning. Le cylindre est montré dans trois poses différentes pour un angle de 0, $\frac{\pi}{4}$ et $\frac{\pi}{2}$. La coloration de la surface indique visuellement la dépendance des poids aux joints. Ainsi la coloration bleutée colore les poids liés au premier joint, alors que la coloration orangé indique les poids liés au dernier joint.

Le calcul est réalisé sur des cylindres composés de 625 sommets. Les erreurs relatives sur le volume sont répertoriées pour différents angles dans le tableau 4.1
On notera qu'en l'absence de correction, la variation de volume est relativement importante

angle (°)	0	10	30	50	70	90
erreur avant correction (%)	0	0.2	1.5	4	7	11
erreur après correction (linéarisé) (%)	0	8.10^{-6}	5.10^{-2}	0.04	0.14	0.34
erreur après correction (exacte) (%)	0	7.10^{-5}	3.10^{-4}	7.10^{-5}	5.10^{-5}	8.10−6

Tab. 4.1: Erreurs relatives du volume du cylindre déformé pour différents angles. On sépare le cas de la correction linéarisée et celle obtenue de façon exacte.

(11%) d'erreur pour un angle de 90°, est déjà corrigé avec une bonne précision dans le cas approximé par la linéarisation. En effet, l'erreur est inférieure à 1% même dans le pire des cas. La correction dans le cas non approximé est quant à elle exacte aux erreurs numériques près.

Rapidité de calcul

On peut constater que l'application de la correction linéarisée ne nécessite qu'une boucle sur l'ensemble des triangles pour le calcul de chaque vecteur de correction. De plus, on constate que la connaissance de la matrice complète des β_{ijk} n'est pas nécessaire. En effet, cette matrice de dimension N_s^3 serait bien trop grosse à stocker (bien que des méthodes puissent être mises en oeuvre pour utiliser l'avantage des matrices creuses). Afin de calculer les vecteurs $\mathbf{V}^{YZ}, \mathbf{V}^{XZ}$ et $\mathbf{V}^{XY}$, on réalise simplement un parcours sur l'ensemble des triangles. On additionne alors

les valeurs correspondantes dans un vecteur de dimension N_s, et aucune boucle supplémentaire n'est nécessaire.

Enfin, les trois vecteurs peuvent être calculés dans la même boucle de parcours des sommets. Les corrections sur les axes x, y et z sont donc réalisées en parallèle. La correction doit ensuite être appliquée pour chaque sommet et nécessite donc un second parcours.

Ainsi la mise en place de la correction linéaire est donc quasiment aussi rapide que la simple connaissance du volume lui-même.

Dans le cas exact, par contre, trois minimisations sont nécessaires à la chaîne. Pour chacune d'elles, une nouvelle boucle de parcours des triangles est nécessaire. Ainsi, la mise en place de la correction exacte nécessite trois fois plus de calcul que la simple connaissance du volume.

Cependant la complexité reste inchangée en $\mathcal{O}(N_s)$, et la mise en place en temps réel est tout à fait envisageable.

Une implémentation non optimisée (allocation à chaque intervalle de temps) en temps-réel sous OpenGL est mise en place comme le montre la figure 4.5. La perte de rapidité d'affichage dans le cas linéarisée et exacte par rapport au simple calcul du volume est tout à fait acceptable, en prenant l'exemple d'un affichage original de 74 fps pour un cylindre composé de 225 sommets, on obtient une correction à la cadence de 53 et 45 fps pour le cas linéarisée et exacte.

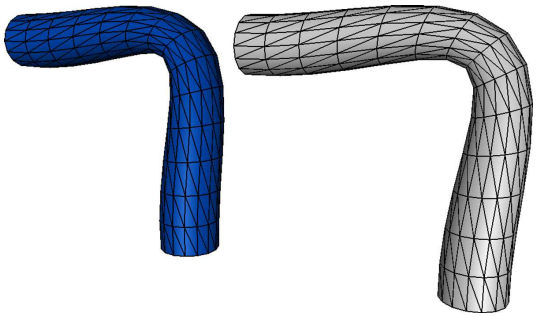


Fig. 4.5: Exemple de visualisation temps-réel sous OpenGL du cylindre déformé par skinning (cylindre gris) ainsi que celui corrigé en volume (cylindre bleu).

Résultats graphique

Les résultats graphiques du cylindre corrigé en volume sont montrées en Fig. 4.6. On remar-

Fig. 4.6: Visualisation du résultat de la correction de volume pour un angle de 90°. Le cylindre de gauche correspond au cas obtenu par simple skinning, alors que le cylindre de droite correspond à celui obtenu après correction de volume.

quera que visuellement, le cas linéarisé et le cas de correction exacte sont tout à fait similaires. Ainsi, le résultat visuel ne correspond à priori pas forcément à celui attendu. En effet, le creux

du joint est toujours présent. Le problème de collapsing elbow n'est donc pas résolu. Le volume est en effet notamment compensé sur les deux extrémités plus épaisses qui se gonflent au fur et à mesure de la rotation.

Les composantes des vecteurs de la correction encodées en tant que couleur sur le cylindre sont montrés en Fig. 4.7 On peut alors noter que les déformations se répartissent uniformément

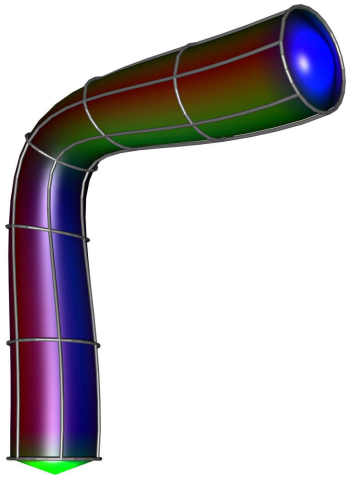


Fig. 4.7: Encodage couleur des composantes de la correction. La coloration rouge correspond à une correction suivant x, la coloration bleue correspond à celle suivant z, et la verte à celle suivant y.

le long du cylindre sur les trois axes. De plus, on peut constater que les deux extrémités du cylindre sont également déplacées de façon à gagner du volume et ce, de façon importante.

4.4.2 Discussion

Cette première étape de correction de volume est donc facile à implémenter et extrêmement rapide à mettre en oeuvre. En effet, cette méthode ne ralentit pas plus l'animation que le simple fait de calculer le volume. Elle possède de plus l'avantage de pouvoir corriger le volume de façon exacte sans surcoût important, bien qu'une version linéarisée soit disponible dans le cas où la recherche d'une vitesse optimale est possible.

Cependant, l'aspect visuel du résultat corrigé n'est pas forcément celui souhaité. En effet, la perte de volume gênante dans le coude du cylindre est compensée de façon uniforme sur le reste du cylindre en gonflant celui-ci dans les trois directions. De plus les extrémités du cylindre que l'on souhaiterait fixes sont également déplacées de façon à compenser le volume.

Ces problèmes proviennent du choix même de la minimisation. En effet, il est moins "coûteux" de déplacer les sommets des régions extrêmes du cylindre en terme de norme du vecteur déplacement $\mathbf{u}$ que ceux du coude pour un même gain en volume.

Enfin, on notera que la correction ne tient absolument pas compte des paramètres du skinning. Ainsi, il s'agit simplement d'une correction agissant sur une surface triangulée quelconque. Cela est évidement une limitation car on ne prend pas en compte la structure organisée de la surface de par la présence de son squelette d'animation.

Il est donc essentiel de corriger plusieurs limitations majeures par la suite

- La correction n'est pas locale mais se répartit de façon homogène sur l'ensemble du cylindre.
  Ce problème est évidemment dû à la prise en compte de l'ensemble des sommets de façon identique dans la minimisation. Afin de pallier à ce problème, on pourrait suggérer la séparation de certaines zones de la surface sur lesquelles on effectuerait la correction de volume indépendamment.

Il serait alors nécessaire de définir ces sections et de pouvoir les raccorder entre elles tout en préservant le degré de continuité désiré pour l'ensemble.
Une autre façon de limiter ce problème est de considérer une pondération de la déformation pour chaque sommet. Ainsi, certains d'entre eux pourraient être plus ou moins faciles à déformer suivant leur position. On peut finalement penser à faire dépendre cette pondération des poids de skinning afin de relier la correction de volume au cas des surfaces déformées par cette méthode.

– La correction a lieu de façon égale sur les trois axes x, y et z.
En effet, cette contrainte a été mise en place de façon arbitraire en égalisant la contribution de la correction selon les trois axes. Il est bien évident que l'on peut modifier la condition, bien qu'il ne soit pas facile de connaitre les contributions de chaque axe pour une pose donnée. En effet, pour cet exemple du cylindre, il parait peu probable que la correction du tiers du volume selon l'axe vertical soit justifiée. En effet, le coude du cylindre nécessiterait quant à lui une correction plus poussée au niveau de l'axe z.
Ce problème provient de la manière même dont la minimisation est abordée. En effet, afin d'utiliser la trilinéarité de l'expression du volume, il a été nécessaire de traiter séparément les axes x, y et z. Cela nous a permis d'obtenir une correction exacte. Par contre, elle empêche la prise en compte d'une déformation au niveau vectoriel dépendante de la forme même de l'objet. Ainsi, on peut supposer qu'une direction de correction plus "naturelle" soit donnée par la direction normale à la surface. Cela permet de plus d'avoir une direction de déformation non plus homogène sur tout l'objet mais locale et dépendant de la forme de celui-ci. Par contre, l'avantage de la séparation trilinéaire sera perdu.
– Certains sommets ne sont pas fixes.
On peut en effet remarquer que les deux sommets extrêmes des cylindres sont déplacés de façon visible pour compenser le volume de celui-ci. Or ces sommets ne devraient pas rentrer dans cette compensation. De même, si l'on découpe le cylindre en différentes sections, il pourrait être avantageux de pouvoir fixer les sommets des deux bordures de façon à assurer la continuité de la surface. Il sera donc également nécessaire de pouvoir définir des contraintes de positions fixes aux niveaux de certains sommets.

4.5 Pondération locale de la déformation

Nous avons donc pu constater que les méthodes précédentes permettent d'obtenir une correction exacte du volume. Cependant, la déformation est indépendante du squelette et des poids de skinning. Cela est évidemment un inconvénient pour le contrôle de la correction car on ne tient pas compte de la structure du caractère animé. Ainsi, la correction de volume se répartit uniformément sur l'ensemble de l'objet.

On pourrait cependant chercher à obtenir une correction de volume locale sur l'objet. Par exemple, dans le cas du cylindre, on pourrait souhaiter que le volume soit conservé dans la région du coude creusé par le skinning, alors les extrémités proches du premier et dernier joints ne devraient pas être modifiées.

Afin de pallier à ce problème, on peut tout d'abord suggérer que les endroits nécessitant une correction soient ceux situés entre deux joints ou plus. En effet, ce sont pour ces sommets que les influences de joints opposés vont avoir tendance à déformer le maillage. La perte de volume peut notamment bien se visualiser dans le cas du "collapsing elbow" où le problème se situe aux articulations. Au contraire, les endroits très influencés par un seul joint vont simplement suivre la direction de celui-ci sans être déformés (translation ou rotation).

On va donc rechercher à effectuer la correction de préférence aux sections où la méthode de skinning modifie le volume. Ces sections qui sont caractérisées par une dépendance à un minimum de deux joints vont donc avoir des poids de skinning faibles pour l'ensemble des repères dont elles dépendent. On note $\omega^s_{\max}$ le poids de skinning maximum sur l'ensemble des joints dont

dépend le sommet s.

Afin de permettre une modification plus aisée des sommets ayant un faible $\omega_{\max}$ et de restreindre la déformation de ceux qui sont proches d'un joint, on modifie la minimisation initialement donnée dans l'eq. 4.3 en pondérant les sommets

$$\begin{cases} \min & \displaystyle\sum_i \left[\frac{\|\mathbf{u}_i\|^2}{\gamma_i}\right] \\ \text{contraint à} & V(\mathbf{x}+\mathbf{u}) = V^0 \ , \end{cases} \tag{4.9}$$

où γ_i va tendre vers 0^+ pour un sommet ne dépendant quasiment que d'un joint ($\omega^s_{\max}$ proche de un) et vers 1 pour un sommet dépendant de plusieurs ($\omega^s_{\max}$ faible).

Une façon de quantifier γ en fonction des poids ω, peut être de considérer, par exemple

$$\gamma_i = \left[1 - (\omega^i_{\max})^q\right]^p \ ,$$

où p et q permettent d'ajuster la difficulté de déformer les sections ne dépendant que d'un seul joint.

Les exemples de zones de déformations privilégiées sont alors montrées pour les différents cas en Fig. 4.8.

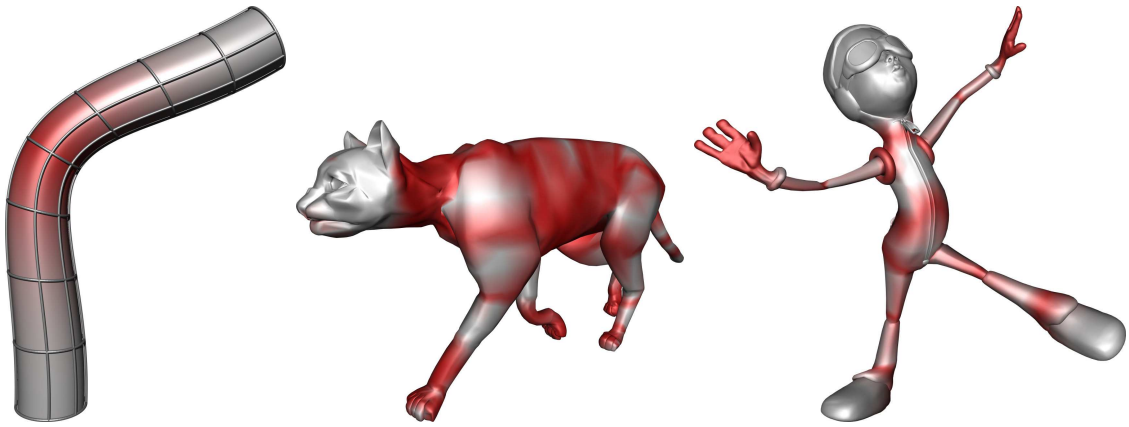


Fig. 4.8: Zones de déformations sur différents exemples. Les zones dépendantes d'au moins deux joints ($\omega_{\max}$ faibles) sont mises en évidences par la coloration rouge.

En reprenant la méthode précédente, on peut réécrire de façon identique les minimisations que l'on recherche à résoudre. Dans le cas de la première déformation selon x, le Lagrangien s'exprime ainsi

$$E_1 = \sum_{i=0}^{N_s} \left[\frac{(u_i^x)^2}{\gamma_i}\right] + \lambda \left[\sum_{(i,j,k)} \beta_{ijk}\, u_i^x y_j z_k - \frac{\Delta V}{3}\right]$$

et de même sur les deux autres axes. On peut alors considérer comme dans la section précédente une résolution linéarisée ou exacte.

Si on note désormais $\mathbf{W}$ tel que

$$\begin{cases} W_i^{YZ} = \gamma_i\, V_i^{YZ} \\ W_i^{XZ} = \gamma_i\, V_i^{XZ} \\ W_i^{XY} = \gamma_i\, V_i^{XY} \ , \end{cases}$$

et en reprenant l'hypothèse que cette correction pondérée se répartit toujours de façon égale sur les trois axes, on peut alors montrer (Annexe C) que le résultat peut se mettre sous la forme

(dans le cas de correction exacte)

$$(\mathbf{u}^x, \mathbf{u}^y, \mathbf{u}^z) = \frac{\Delta V}{3}\left(\frac{\mathbf{W}^{YZ}}{\langle \mathbf{W}^{YZ}, \mathbf{V}^{YZ}\rangle}, \frac{{}^*\mathbf{W}^{XZ}}{\langle {}^*\mathbf{W}^{XZ}, {}^*\mathbf{V}^{XZ}\rangle}, \frac{{}^{**}\mathbf{W}^{XY}}{\langle {}^{**}\mathbf{W}^{XY}, {}^{**}\mathbf{V}^{XY}\rangle}\right) . \tag{4.10}$$

On remarquera que la structure de la solution reste identique bien que deux vecteurs de la solution soient légèrement décalés dans l'espace $\mathbb{R}^{N_s}$.

4.5.1 Résultats

Les résultats obtenus pour l'exemple du cylindre dans le cas où $p = 8$ et $q = 15$ sont montrés en Fig. 4.9.

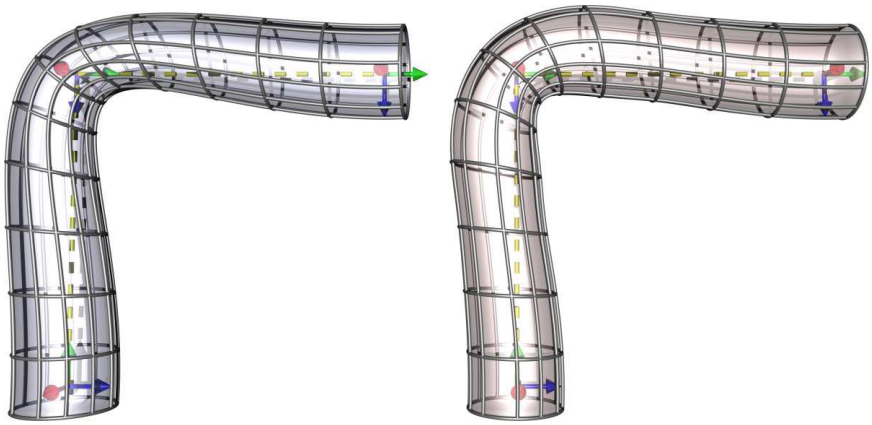


Fig. 4.9: Résultats obtenus pour le cas des sommets pondérés suivant les poids de skinning max pour un angle de $\frac{\pi}{2}$.

On peut maintenant constater que la déformation est cette fois bien limitée à l'intérieur du coude du cylindre. Celui-ci ne subit plus la diminution locale du volume gênante visuellement et permet donc d'améliorer l'aspect de l'animation.

Les résultats numériques de la contrainte de volume sont identiques au cas non pondéré et l'implémentation ne nécessite que peu de modifications par rapport à la méthode précédente. La rapidité de l'exécution de la contrainte de volume est donc gardée intacte.

On peut également visualiser la répartition de la correction encodé en couleurs comme dans le cas précédent en Fig. 4.10. On remarque cette fois clairement que celle-ci a bien lieu dans l'entourage du coude et non plus sur les bords extrêmes du cylindre.

Enfin, on peut comparer l'aspect visuel du cylindre corrigé pour différentes valeurs de p et q. Lorsque q augmente, la déformation va être de plus en plus localisée dans le coude autour du minimum de $\omega_{\max}$. Et au contraire, lorsque p augmente, la répartition de la déformation va se lisser le long du cylindre. Le cas où l'on considère $p = q$ donne des résultats graphiques corrects. Des affichages sont montrés pour certaines valeurs en Fig. 4.11.

On peut remarquer que pour une valeur faible de p et q, la pondération est peu ciblée, et on observe le même type de correction que dans le cas homogène avec une perte de volume locale au niveau du coude. Au contraire, lorsque p et q augmentent, le coude compense localement son volume et le résultat visuel s'améliore. Pour des valeurs intermédiaires une légère non dérivabilité est induite au niveau du coude alors que pour des valeurs importantes, le coude semble suivre une trajectoire quasi circulaire avec un rayon constant, bien qu'un point d'inflexion se dessine nettement avant le passage sur le coude.

Dans le cas où $p \neq q$, on peut observer d'autres types de comportements. Ainsi si q est trop important, la déformation est très localisée et pas assez lissée par le paramètre p, les sommets situés au milieu du coude vont compenser à eux seuls une valeur importante du volume et

Fig. 4.10: Codage couleur des déplacements suivant leur direction et leur amplitude pour le cas de la correction pondérée. L'exemple est pris pour p=8 et q=15.

Fig. 4.11: Influence des paramètres p et q sur l'aspect du cylindre déformé. Les valeurs prises sont respectivement de $p = q = 0.2$, 5, 10 et 15 de gauche à droite et de haut en bas.

déformer considérablement la surface. Au contraire, si la déformation est localisée mais également très lissée par p, on observe une sorte de boule se former à la jonction compensant à elle seule la perte de volume.

4.5.2 Discussion

La mise en place de la pondération des sommets permettant de faciliter ou non la correction de leur position permet donc d'améliorer de façon très importante l'aspect graphique du résultat. Cette correction permet en effet de tenir compte de la structure du caractère de par la connaissance des zones plus ou moins déformées lors du skinning. Elle permet notamment de réduire l'effet du "collapsing elbow" pour un surcoût de calcul faible.

Le choix des pondérations considéré possède l'avantage de pouvoir être calculé automatique-

Fig. 4.12: Cas extrêmes de la déformation très localisée. La figure de gauche montre le cas où la déformation est trop localisée (q=10) et pas assez lissée (p=1), alors que celle de droite montre une déformation trop localisée (q=10) et très lissée (p=30).

ment d'après l'information des poids de skinning seule. Ainsi, aucun travail supplémentaire n'est nécessaire à l'artiste pour définir des zones de conservation de volume. On peut également noter que ces pondérations possèdent deux degrés de liberté d'après la formule utilisée. Ce qui permet d'ajuster la forme désirée. On remarquera cependant que pour des valeurs de p et q proches de l'unité ou inférieures, un problème de non dérivabilité intervient au milieu du coude. La surface tend donc à être simplement C^0. Ce problème provient de l'utilisation dans la formulation du paramètre $\omega_{\max}$ qui est évidement non dérivable à la jonction. L'utilisation de la fonction max qui n'est pas continue pose donc un problème pour des surfaces lisses. Pour cela, on a pu noter que l'élévation du degré des paramètres permet de gommer cet effet en lissant la répartition des pondérations.
Cependant, on pourrait également envisager de modifier la définition des γ_i afin de ne plus utiliser de fonction max. Il suffirait pour cela de considérer une fonction continue et dérivable avantageant nettement les valeurs supérieures à 0.5 et désavantageant les autres. Cette fonction ne nécessite d'être calculée qu'une seule fois (les γ_i pouvant être stockés pour l'ensemble des itérations), son implémentation peut utiliser des fonctions non linéaires sans ralentir l'exécution du programme.

De plus, on remarque que le lien entre les pondérations et les paramètres du skinning peut être modifié. On peut ainsi considérer l'exemple d'un bras pour lequel le muscle gonfle lors du mouvement. Il est alors envisageable de définir une région sur la surface correspondant à la présence du muscle. Cette région pourrait alors combler en partie la différence de volume et ainsi gonfler lors de la déformation. De même, il pourrait être possible de définir certaines régions adipeuses de cette façon.

La mise en place de cette méthode possède également l'avantage d'avoir quasiment supprimé le problème des deux sommets extrêmes du cylindre qui ne sont quasiment plus déformés par la pondération choisie. Cependant ce problème est minimisé par le choix des points mais n'est pas totalement éliminé. De même, si l'on découpe le cylindre en plusieurs bouts, il pourrait toujours être intéressant d'avoir la possibilité de fixer certaines positions.

4.6 Contrainte de position fixe

La méthode précédente ne considère pas de contraintes au niveau de certaines positions. Ainsi l'algorithme risque de déplacer des sommets que l'on souhaite fixes. On a pu notamment voir que pour le cas du cylindre, les deux sommets extrema ne servant qu'à refermer la surface pouvaient poser certains problèmes. Ces deux sommets n'ont aucun intérêt à être déplacés lors de la contrainte de volume car ils ne jouent pas de rôle sur l'aspect extérieur de la surface. De

même, il peut être intéressant de contraindre certains sommets comme les bords du cylindre afin de réaliser des jonctions morceaux à morceaux.

Dans ce cas, on peut rajouter une contrainte supplémentaire au niveau de chaque sommet. On suppose que les indices des sommets non fixes sont situés dans l'ensemble $\mathcal{F}$. On peut donc écrire la contrainte supplémentaire par

$$\forall i \notin \mathcal{F},\ \|\mathbf{u}_i\| = 0\ ,$$

où bien, de façon identique

$$\sum_{i \notin \mathcal{F}} \|\mathbf{u}_i\|^2 = 0\ .$$

La résolution se déroule de la même façon en prenant toujours la méthode des multiplicateurs de Lagrange, avec, cette fois, deux contraintes. La résolution est détaillée en Annexe D.

On dénote maintenant $\mathbf{a}_{|\mathcal{F}}$ le vecteur dont les composantes sont données par $a_{i_{|\mathcal{F}}} = a_i$ si $i \in \mathcal{F}$ et valant 0 sinon. Cette solution s'exprime alors simplement sous la forme

$$(\mathbf{u}^x, \mathbf{u}^y, \mathbf{u}^z) = \frac{\Delta V}{3}\left(\frac{\mathbf{W}^{YZ}_{|\mathcal{F}}}{\left\langle \mathbf{W}^{YZ}_{|\mathcal{F}}, \mathbf{V}^{YZ}_{|\mathcal{F}}\right\rangle}, \frac{{}^*\mathbf{W}^{XZ}_{|\mathcal{F}}}{\left\langle {}^*\mathbf{W}^{XZ}_{|\mathcal{F}}, {}^*\mathbf{V}^{XZ}_{|\mathcal{F}}\right\rangle}, \frac{{}^{**}\mathbf{W}^{XY}_{|\mathcal{F}}}{\left\langle {}^{**}\mathbf{W}^{XY}_{|\mathcal{F}}, {}^{**}\mathbf{V}^{XY}_{|\mathcal{F}}\right\rangle}\right)\ . \tag{4.11}$$

La structure est donc identique à la méthode précédente à la différence près que les vecteurs $\mathbf{V}$ et $\mathbf{W}$ sont cette fois considérés dans un espace de dimension plus faible.

4.7 Direction Normale

Les méthodes précédentes sont limitées de par le fait que l'on ne possède pas de contrôle sur la direction de la déformation. Ainsi, il peut être préférable de fixer une direction locale pour chaque sommet suivant lequel le déplacement est contraint.

Une direction "naturelle" pour la déformation consiste à fixer des déplacements normaux à la surface. On suppose donc que l'on connait pour une pose donnée du skinning les normales unitaires $\mathbf{n}_i$ de chaque sommet i.

Cette fois, la contrainte de volume s'exprime alors par

$$\begin{cases} \min \displaystyle\sum_{i=0}^{N_s} \frac{\rho_i^2}{\gamma_i} \\ \\ \text{contraint à } V(\mathbf{x}+\boldsymbol{\rho}) = V^0 \end{cases} \tag{4.12}$$

où $\boldsymbol{\rho}$ représente le déplacement dans la direction normale au sommet donné ($\boldsymbol{\rho}_i = \rho_i \mathbf{n}_i$).

Cas linérarisé

On exprime cette fois le cas linéarisé de cette déformation au niveau de la contrainte de volume.

En négligeant les termes quadratiques et le terme cubique on peut alors écrire la minimisation de l'eq. 4.12 par

$$\begin{cases} \min & \displaystyle\sum_i \frac{\rho_i^2}{\gamma_i} \\ \\ \text{contraint à} & V(\boldsymbol{\rho}^x, \mathbf{y}, \mathbf{z}) + V(\mathbf{x}, \boldsymbol{\rho}^y, \mathbf{z}) \\ & +V(\mathbf{x}, \mathbf{y}, \boldsymbol{\rho}^z) = \Delta V \end{cases} \tag{4.13}$$

De la même façon que précédemment, on peut exprimer le lagrangien de l'expression par

$$E = \sum_{i=0}^{N_s} \frac{\rho_i^2}{\gamma_i} + \lambda \left[\sum_{(i,j,k)} \beta_{ijk} \Big(\rho_i n_i^x y_i z_k + x_i \rho_j n_j^y z_k + x_i y_j \rho_k n_k^z \Big) - \Delta V \right]$$

On peut alors montrer (Annexe E) que la résolution du système $\left(\frac{\partial E}{\partial \rho_i} = 0, \frac{\partial E}{\partial \lambda} = 0\right)$ fournit l'expression du déplacement suivante

$$\rho_i = \Delta V \frac{\gamma_i < \mathbf{n}_i, \mathbf{v}_i >}{\sum_{i'} \gamma_{i'} < \mathbf{n}_{i'}, \mathbf{v}_{i'} >^2} , \tag{4.14}$$

où l'on utilise la notation $\mathbf{v}_i = (V_i^{YZ}, V_i^{XZ}, V_i^{YZ})$.

Résultats

Cette fois la déformation n'est plus limitée à se réaliser forcément dans les trois directions de l'espace par un facteur prédéfini à l'avance. Celle-ci s'adapte donc automatiquement à la forme du personnage animé.

Cependant, on notera en Fig. 4.13 que la déformation obtenue ne donne plus le même effet sur la surface corrigée. Ainsi dans le cas du cylindre, ce n'est pas le bord intérieur du coude qui va se déformer pour contrer la perte de volume, mais ce sont, au contraire, les régions latérales de ce coude qui vont s'étirer.

On peut également observer l'amplitude des déplacements de la correction encodée en couleur en Fig. 4.14. On remarque effectivement l'aspect très localisé de la correction au niveau des côtés du cylindre.

Enfin, on notera que les résultats numériques donnent une erreur relative sur le volume inférieur à 5% après correction compte tenue de la linéarisation de la contrainte. Les erreurs relatives du volume pour différents angles sont ainsi répertoriées sur le tableau 4.2.

angle (°)	0	10	30	50	70	90
erreur avant correction (%)	0	0.2	1.5	4	7	11
erreur après correction dans la direction normale(linéarisé) (%)	0	0.05	0.5	1.5	3.1	4.9

Tab. 4.2: Erreur relative du volume du cylindre déformé pour différents angles et un cylindre composé de 256 sommets.

Discussion

Cet effet de déformation dans la direction transverse du creux du coude n'est pas forcément celui attendu initialement mais correspond cependant à une déformation tout à fait plausible d'un point de vue physique. Cet effet peut être mis en parallèle avec l'aspect qu'aurait un tube de caoutchouc que l'on plierait. Il peut ainsi être tout à fait intéressant d'appliquer cette contrainte dans le cas d'animation de jouets en plastique élastique.

Cet effet s'explique au niveau de la résolution par le simple fait qu'il est moins coûteux en terme de norme du vecteur de correction **u** de déplacer les sommets situés sur les bords latéraux que ceux de l'intérieur du coude pour un même gain en terme de volume. En effet, de par la perte de volume engendré dans le creux du coude suite à l'application du skinning classique, les sommets intérieurs sont plus proches les uns des autres. Il en résulte donc que le déplacement de l'un de ces sommets ne modifie que peu le volume considéré. Ce phénomène n'apparaissait pas

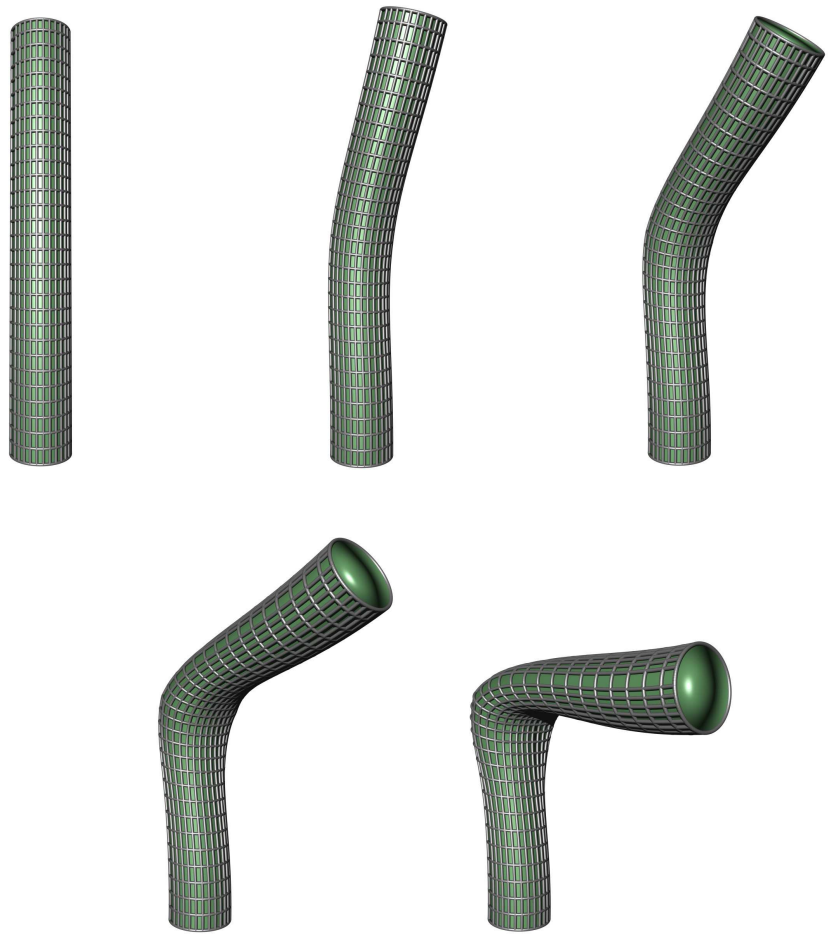


Fig. 4.13: Résultats obtenues pour le cas d'une correction normale à la surface. Le cylindre est déformé pour des angles de plus en plus importants.

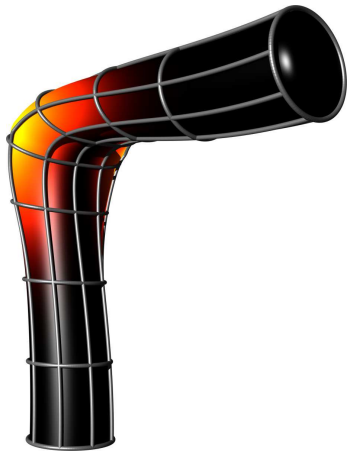


Fig. 4.14: Encodage couleur de l'amplitude de la correction suivant la direction normale. Cette fois, les teintes des couleurs n'indiquent plus de direction particulière car seule l'amplitude de la déformation importe.

dans la méthode précédente car l'on forçait chacun des axes à absorber un tiers de la variation de volume.

Néanmoins, il est tout à fait possible d'envisager certaines modifications sur les paramètres de la minimisation pour redonner l'effet initial de correction de volume appliqué principalement

dans le coude. On peut ainsi citer par exemple deux types de solutions devant permettre de donner cet effet

- Définir la région de déformation de façon non radiale.
  En définissant les régions à déformer uniquement dans le creux du coude où l'on désire la compensation de volume, on peut ainsi empêcher la création de la compensation latérale. Cependant, on perd l'intérêt du calcul automatique des régions à déformer d'après l'information des poids de skinning seuls.
- Forcer la correction à limiter les déformations de voisinages par rapport à la forme skinnée.
  En effet si la forme skinnée peut être considérée comme correcte au sens du voisinage, on va vouloir limiter non seulement la déformation au niveau de l'amplitude des vecteurs de déplacement, mais également au niveau de la déformation du voisinage. On peut notamment penser à introduire des systèmes masses-ressorts au modèle. Cependant la non linéarité de la contrainte à résoudre empêche sa mise en place en temps réel. Le cas des coordonnées laplaciennes est une autre approche qui possède l'avantage de pouvoir s'exprimer de façon linéaire. Il peut ainsi être avantageux de minimiser également la déformation dans cet espace de coordonnées.

4.8 Ajout des coordonnées laplaciennes

Mise en place du problème

On rapelle que les coordonnées laplaciennes sont définies pour un sommet i par

$$\mathcal{L}(\mathbf{x}_i) = \frac{1}{\mathcal{N}_i} \sum_{j \in \mathcal{V}_i} (\mathbf{x}_i - \mathbf{x}_j) \ ,$$

où $\mathcal{N}_i$ représente le nombre de voisins du sommet $\mathbf{x}_i$ et $\mathcal{V}_i$ sont les indices des sommets voisins.

Il peut maintenant être intéressant de limiter la correction appliquée dans ce système de coordonnées. On va donc chercher à minimiser l'écart entre les coordonnées laplaciennes déformées et celles issues directement du skinning.

$$\sum_i \frac{1}{\gamma_i} \left\| \mathcal{L}(\mathbf{x}_i + \rho \mathbf{n}) - \mathcal{L}(\mathbf{x}_i) \right\|^2 \ .$$

Or on peut noter que l'opérateur $\mathcal{L}$ des coordonnées laplaciennes est linéaire, l'expression se résume donc à

$$\sum_i \frac{1}{\gamma_i} \left\| \mathcal{L}(\rho \mathbf{n}) \right\|^2 \ .$$

Pour cela, une nouvelle minimisation peut s'écrire sous la forme

$$\begin{cases} \min & \displaystyle\sum_i \frac{(\rho_i)^2}{\gamma_i} + \kappa \sum_i \frac{1}{\gamma_i} \left\| \mathcal{L}(\rho\, \mathbf{n}_i) \right\|^2 \\ \text{contraint à} & V(\mathbf{x} + \rho\, \mathbf{n}) - V^0 = 0 \ , \end{cases}$$

où κ représente un simple facteur de poids entre la minimisation dans l'espace cartésien et celle dans l'espace laplacien.

On remarquera que les coordonnées laplaciennes se rapprochent de l'expression de la courbure de la surface (expression approchée uniquement). Cette minimisation peut donc être mise en parallèle avec une certaine minimisation d'énergie au niveau de la surface corrigée.

Expression de la solution

On peut montrer (Annexe F) que la solution linéarisée au premier ordre au niveau de la contrainte de volume peut s'écrire

$$\forall i\,,\ \rho_i = \Delta V \frac{q_i}{\sum_j q_j \ < \mathbf{n}_j, \mathbf{v}_j >}\ ,$$

$\mathbf{q}$ étant le vecteur formé par le produit matriciel $\mathrm{A}^{-1}\,\mathbf{K}$. $\mathbf{K}$ étant le vecteur formé par les K_i tels que $K_i =< \mathbf{n}_i, \mathbf{v}_i >$, où l'expression des $\mathbf{v}_i$ a été introduite dans la section précédente. Et A est la matrice définie par les coefficients

$$\begin{aligned}\mathrm{A}_{ij} \quad &= \frac{2}{\gamma_i}\left(1+\kappa\|\mathbf{n}_i\|^2\right)\\ &+ 2\kappa\left[\sum_k \frac{\delta_k^i}{\mathcal{N}_k\gamma_k}\|\mathbf{n}_i\|^2\right.\\ &\left.- \left\langle \mathbf{n}_i, \left(\frac{\delta_{\mathcal{V}_i}^j}{\mathcal{N}_i\,\gamma_i} + \frac{\delta_{\mathcal{V}_j}^i}{\mathcal{N}_j\,\gamma_j}\right)\mathbf{n}_j\right\rangle\right]\ ,\end{aligned}$$

et les $\delta_{\mathcal{V}_j}^i$ vallent un si $i \in \mathcal{V}_j$ et zéro sinon.

On remarquera par contre que cette fois, un système linéaire doit être résolu à chaque itération ce qui ralentit sa mise en place. On notera cependant que cette matrice à inverser est très creuse, car issue d'opérateurs de voisinage directs. De plus, seule l'expression du produit $\mathrm{A}^{-1}\,\mathbf{K}$ est nécessaire, la connaissance des valeurs de A^{-1} n'étant pas nécessaire. Différents algorithmes itératifs rapides peuvent donc être utilisés afin de résoudre ce système le plus rapidement possible.

Cette méthode n'a cependant pas encore été implémentée dans l'état actuel de ce travail.

Chapitre 5

Travaux Futurs

Différentes méthodes de correction ont donc pu être testées. On a pu constater que des résultats satisfaisants pour le modèle du cylindre étaient obtenus dès que l'on prenait en compte les paramètres du skinning afin de pondérer la déformation. La mise en place de la déformation suivant la normale permet de plus d'obtenir une déformation robuste par rapport à la nature plus "physique" de la direction de cette correction. De plus cette méthode a pu être implémentée en *C++* sans ralentissements importants de l'affichage pour les modèles considérés.

Validation sur d'autres modèles.

Nous souhaiterions par la suite pouvoir valider les résultats sur des modèles plus complexes que celui du cylindre. Nous pensons notamment utiliser le modèle du chat afin de vérifier que la méthode apporte une amélioration visuelle notable lors d'un mouvement défini par l'artiste. Un exemple est ainsi montré en Fig. 5.1 dans le cas du chat pour une correction localisée sur la patte.

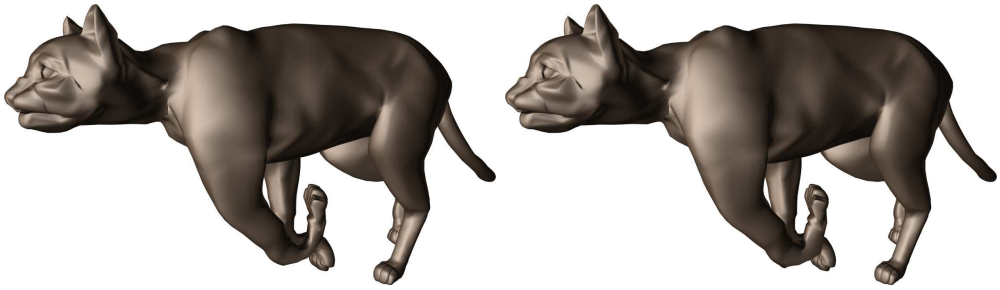


Fig. 5.1: Application de contrainte de volume sur l'exemple du modèle de chat. La figure de gauche montre le cas du skinning simple qui déforme la patte avant gauche en réduisant considérablement son volume. La figure de droite montre l'effet de la correction pondérée dans le cas où le volume est corrigé de façon égale sur les trois axes. La correction de volume est également pondérée de sorte à restreindre la correction au niveau de cette patte avant gauche.

Choix des poids de pondération.

Nous avons également pu constater que le choix des poids de pondération permettait de faire varier l'effet obtenu sur la surface finale. Nous souhaiterions donc étendre le choix du calcul des poids de pondération avec la possibilité de ne pas utiliser uniquement le paramètre $\omega_{\max}$ sur les poids de skinning.

Il sera donc possible de mettre en place de façon manuelle une région de type muscle définie non radialement sur un exemple de bras. Ainsi, nous pensons qu'il sera possible de donner

simplement l'impression du gonflement du muscle lié à cette contrainte sur le volume.

De plus, il est tout à fait possible de considérer le problème d'une autre façon en définissant cette fois les régions à déformer non plus comme celles étant les plus éloignées des joints, mais par des zones intermédiaires. En effet, en considérant l'exemple d'un doigt qui se plie, ce sont les parties adipeuses situées entre les joints qui gagnent du volume. Nous tenterons donc de définir les futures régions à déformer comme les régions intermédiaires situées de part et d'autre du coude de la jonction afin de modéliser ce type de comportement. L'avantage de cette approche est qu'elle ne nécessitera pas non plus de définition manuelle de ces régions. En effet, les pondérations pourront toujours être calculées de façon automatique suivant les valeurs des poids de skinning.

Enfin, nous souhaiterions introduire un paramètre supplémentaire intrinsèque à l'animation par squelette dans le choix des poids, à savoir la distance entre les sommets et le squelette. En effet, au sens anatomique, les régions les plus éloignées du squelette sont généralement les régions adipeuses se déformant le plus facilement. Or les squelettes d'animation sont généralement relativement proches des squelettes anatomiques [RFDC05]. Il est donc possible de définir ces régions à déformer de façon plus précise grâce à cette information de distance. On pourra également noter que cette nouvelle information est porteuse de l'aspect non radial de la correction à appliquer. La mise en place de sections adipeuses liées à la distance du squelette a notamment déjà été utilisée pour la mise en place d'oscillations de ces régions lors d'un mouvement rapide [Lar04].

Extension aux domaines d'animation de jouets plastiques.

Nous avons pu constater que le choix de la fonction à minimiser permettait de donner différentes caractéristiques notables visuellement sur la surface corrigée. Nous avons également pu constater que la prise en compte de la déformation normale à la surface donnait naturellement un aspect physiquement plausible pour des objets de type caoutchouc. Nous souhaiterions ainsi étendre notre recherche non seulement aux cas de personnages "vivants" mais également aux modèles de type jouets en plastique déformable, rendant compte de ce type de comportement.

Mise en place de la prise en compte du voisinage.

La dernière partie des résultats de ce travail nous donne une solution permettant de prendre en compte le voisinage des sommets de la surface. Il devient donc possible de prendre en compte une minimisation au niveau de l' "énergie" de la surface elle-même et ainsi de se rapprocher d'un comportement physiquement réaliste de certaines déformations. Cela permet de plus de rendre les surfaces plus lisses en minimisant différents types de discontinuités.

La solution a pu être explicitée et nous souhaitons donc mettre en place son implémentation dans une optique temps réel.

Localité de la correction.

Jusqu'à présent, nous sommes intéressés au problème de la variation globale de volume du domaine bordé par la surface complète. Dans le cas de modèles plus compliqués (personnage ou animal entier), il est nécessaire de séparer les différentes parties du corps afin de ne corriger le volume que sur les parties concernées.

Il est évidemment possible de séparer ces parties manuellement afin de rendre la correction locale. Nous souhaiterions cependant étudier la possibilité de rendre cette segmentation automatique en se basant sur les régions définies par un ou plusieurs joints suivant les valeurs des poids de skinning (et prise en compte de la distance au squelette).

Les sections pouvant alors être accolées les unes au autres en assurant une continuité au minimum C^0 en prenant en compte l'application de la contrainte de positions fixes d'ores et déjà établie en section 4.6.

Correction en une seule étape.

Enfin, on peut noter que les différentes méthodes mises en place agissent sur l'objet en deux étapes. Une étape de skinning classique suivie d'une étape de correction.

Nous pensons également envisager la recherche d'une correction directement appliquée dans le calcul du skinning lui même. Pour cela, il peut être intéressant de modifier au cours du temps certains paramètres comme les poids de skinning de façon à ce que la surface garde un volume constant lors de sa déformation.

Chapitre 6

Conclusion

Une méthode de déformation de surface par la méthode de skinning sous la contrainte d'un volume constant a donc pu être établie. Cette méthode ne nécessite que les informations de base de l'animation par skinning (squelette et poids de skinning) et peut donc s'appliquer directement à de nombreux modèles de l'industrie du jeu vidéo et du cinéma sans nécessiter plus de traitement de la part d'un artiste. La méthode liée au paramètres du skinning peut s'appliquer dans le cadre d'animation interactive ou temps-réel de par le faible coût de l'algorithme de correction en termes de calcul qui est aussi rapide que le simple calcul du volume du personnage.

Plusieurs applications de ces méthodes peuvent être envisagées. Une méthode très simple de calcul approché permet de corriger le volume à moindre coût pour une recherche de rapidité d'exécution optimale. Une méthode exacte permet de contraindre le volume sans surcoût majeur si la valeur numérique de celui-ci doit absolument être respectée. Enfin des méthodes approchées au niveau du respect du volume mais plus robuste d'un point de vue physique peuvent également être mises en place par la prise en compte de la correction dans la direction normale de l'objet considéré.

Nous pensons poursuivre l'approfondissement de ces méthodes en les mettant en place sur des modèles plus complexes d'humanoïdes et d'animaux complets. De plus nous souhaitons élargir leurs cadres d'application de par le choix d'autres paramètres entrant en compte dans le calcul de la correction. Différent effets visuels pourront alors être choisis suivant la correction appliquée (tissus adipeux, présence d'un muscle, jouet en plastique, ...).

Annexe A

Correction de volume sur les trois axes dans le cas linéarisé

On recherche ici à résoudre la minimisation contrainte donnée en section 4.3.
On rappelle que dans ce cas, la fonctionnelle d'énergie est donnée par

$$E(\mathbf{u},\lambda) = \|\mathbf{u}\|^2 + \lambda \left[< \mathbf{u}^x, \mathbf{V}^{YZ} > + < \mathbf{u}^y, \mathbf{V}^{XZ} > + < \mathbf{u}^z, \mathbf{V}^{XY} > -\Delta V \right] \ .$$

Avec la répartition uniforme de la correction exprimée par

$$\begin{cases} < \mathbf{u}^x, \mathbf{V}^{YZ} >= \dfrac{\Delta V}{3} \\ \\ < \mathbf{u}^y, \mathbf{V}^{XZ} >= \dfrac{\Delta V}{3} \\ \\ < \mathbf{u}^z, \mathbf{V}^{XY} >= \dfrac{\Delta V}{3} \end{cases}$$

On peut remarquer que l'énergie peut s'écrire sous la forme développée au niveau de chaque composante par

$$E(\mathbf{u},\lambda) = \sum_i \left((u_i^x)^2 + (u_i^y)^2 + (u_i^z)^2 \right) + \lambda \left[\sum_{(i,j,k)} \beta_{ijk} \left(u_i^x y_j z_k + x_i u_j^y z_k + x_i y_j u_k^z \right) - \Delta V \right]$$

La première partie du système à résoudre donnée par la dérivation de E par rapport à $\mathbf{u}^x$ fournit alors

$$\forall i \in [\![1, N_s]\!] \ , \quad \frac{\partial E}{\partial u_i^x} = 2\, u_i^x + \lambda \sum_{(j,k)} \beta_{ijk}\, y_j z_k = 0 \ .$$

On peut donc isoler u_i^x pour en tirer

$$u_i^x = -\frac{\lambda}{2} \sum_{(j,k)} \beta_{ijk}\, y_j z_k \ . \tag{A.1}$$

La seconde condition de la résolution du lagrangien est donnée par la mise en place de la répartition du volume. On a donc

$$< \mathbf{u}^x, \mathbf{V}^{YZ} >= \sum_{(i,j,k)} \beta_{ijk}\, u_i^x y_j z_k = \frac{\Delta V}{3} \ .$$

En introduisant l'expression précédente de u_i^x, on obtient alors

$$\sum_{(i,j,k)} \beta_{ijk} \left(-\frac{\lambda}{2} \sum_{(j',k')} \beta_{ij'k'} y_{j'} z_{k'} \right) y_j z_k = \frac{\Delta V}{3}$$

$$\Rightarrow -\frac{\lambda}{2} \left[\sum_i \left(\sum_{(j,k)} \beta_{ijk}\, y_j z_k \right)^2 \right] = \frac{\Delta V}{3}$$

$$\Rightarrow \lambda = -\frac{2\,\Delta V}{3} \frac{1}{\sum_i \left(\sum_{(j,k)} \beta_{ijk}\, y_j z_k \right)^2}$$

Donc finalement, en injectant cette expression dans l'eq. A.1, on obtient

$$\forall i \in [\![1, N_s]\!]\,, \quad u_i^x = \frac{\Delta V}{3} \frac{\displaystyle\sum_{(j,k)} \beta_{ijk}\, y_j z_k}{\displaystyle\sum_{i'} \left(\sum_{(j,k)} \beta_{i'jk}\, y_j z_k \right)^2}$$

Finalement, en utilisant à nouveau les notations utilisées en eq. 3.7, on peut écrire

$$\forall i \in [\![1, N_s]\!]\,, \quad u_i^x = \frac{\Delta V}{3} \frac{V_i^{YZ}}{\displaystyle\sum_j (V_j^{YZ})^2}\ ,$$

ou bien, sous forme vectorielle dans $\mathbb{R}^{N_s}$

$$\mathbf{u}^x = \frac{\Delta V}{3} \frac{\mathbf{V}^{YZ}}{\|\mathbf{V}^{YZ}\|^2}\ ,$$

De façon strictement identique, il est évident que la correction sur les autres axes s'exprime de la même façon. On obtient donc

$$\mathbf{u}^y = \frac{\Delta V}{3} \frac{\mathbf{V}^{XZ}}{\|\mathbf{V}^{XZ}\|^2}\ ,$$

$$\mathbf{u}^z = \frac{\Delta V}{3} \frac{\mathbf{V}^{XY}}{\|\mathbf{V}^{XY}\|^2}\ ,$$

Annexe B

Correction de volume sur les trois axes dans le cas exact

On cherche ici à résoudre la minimisation de la section 4.4 de façon exacte dans le cas où l'on réalise trois corrections à la chaîne sur x, y et z.

B.1 Séparation des étapes

On décompose donc la correction en trois étapes distinctes.

1. La première déformation sur l'axe x est donc telle que l'on corrige un tiers de l'erreur sur le volume.
$$V(\mathbf{x}+\mathbf{u}^x,\mathbf{y},\mathbf{z}) = \frac{2}{3}V + \frac{1}{3}V^0 = V + \frac{\Delta V}{3}\ .$$
On obtient donc en décomposant $V(\mathbf{x}+\mathbf{u}^x,\mathbf{y},\mathbf{z}) = V(\mathbf{x},\mathbf{y},\mathbf{z}) + V(\mathbf{u}^x,\mathbf{y},\mathbf{z})$ la relation identique au cas linéarisé
$$V(\mathbf{u}^x,\mathbf{y},\mathbf{z}) = \frac{\Delta V}{3}\ .$$
La première minimisation que l'on recherche à résoudre est donc donnée par le seul déplacement sur l'axe x et son Lagrangien s'exprime donc par :
$$E_1 = \sum_{i=0}^{N_s}(u_i^x)^2 + \lambda\left[\sum_{(i,j,k)}\beta_{ijk}\ u_i^x y_j z_k - \frac{\Delta V}{3}\right]$$

2. La seconde déformation a lieu sur l'axe y. On corrige cette fois la moité de l'erreur restante, on obtient donc
$$V(\mathbf{x}+\mathbf{u}^x,\mathbf{y}+\mathbf{u}^y,\mathbf{z}) = \frac{1}{3}V + \frac{2}{3}V^0 = V + \frac{2\,\Delta V}{3}\ .$$
De la même façon, on décompose l'expression par bilinéarité pour obtenir
$$V(\mathbf{x},\mathbf{u}^y,\mathbf{z}) + V(\mathbf{u}^x,\mathbf{u}^y,\mathbf{z}) = \frac{\Delta V}{3}$$
Le lagrangien de cette minimisation est donc le suivant
$$\begin{aligned} E_2 &= \sum_{j=0}^{N_s}(u_j^y)^2 \\ &+\lambda\left[\sum_{(i,j,k)}\beta_{ijk}\left((x_i+u_i^x)\,u_j^y z_k\right) - \frac{\Delta V}{3}\right]\end{aligned}$$

3. Enfin, concernant la déformation le long de l'axe z, on obtient évidement de la même façon en recherchant $V(\mathbf{x}+\mathbf{u}^x, \mathbf{y}+\mathbf{u}^y, \mathbf{z}+\mathbf{u}^z) = V^0$,

$$\begin{aligned}&V(\mathbf{x},\mathbf{y},\mathbf{u}^z) + V(\mathbf{u}^x,\mathbf{y},\mathbf{u}^z) + V(\mathbf{x},\mathbf{u}^y,\mathbf{u}^z)\\&+V(\mathbf{u}^x,\mathbf{u}^y,\mathbf{u}^z) = \tfrac{\Delta V}{3}\ .\end{aligned}$$

Cette fois, le Lagrangien correspondant est donc

$$\begin{aligned}E_3 &= \sum_{k=0}^{N_s}(u_k^z)^2\\&+\lambda\left[\sum_{(i,j,k)}\beta_{ijk}\left((x_i+u_i^x)(y_j+u_j^y)u_k^z\right) - \frac{\Delta V}{3}\right]\end{aligned}$$

B.2 Résolution

La première résolution ne pose évidement aucun problème puisqu'elle est identique au cas linéarisé (on corrige un tiers du volume suivant un axe). On peut donc écrire directement la solution sous la forme

$$\mathbf{u}^x = \frac{\Delta V}{3}\frac{\mathbf{V}^{YZ}}{\|\mathbf{V}^{YZ}\|^2}\ .$$

La seconde minimisation se résout de façon identique. En effet, dans le cas de E_2, on cherche à résoudre le système

$$\begin{cases}\forall j,\ 2\,u_j^y + \lambda\displaystyle\sum_{(i,k)}\beta_{ijk}\,(x_i+u_i^x)\,z_k = 0\\[2ex]\displaystyle\sum_{(i,j,k)}\beta_{ijk}\left((x_i+u_i^x)\,u_j^y z_k\right) = \frac{\Delta V}{3}\end{cases}$$

où cette fois u_i^x est maintenant connue de par la première résolution.

En isolant u_j^y dans la première équation, et en l'injectant dans la seconde condition du système, on obtient alors

$$\lambda = -\frac{2\,\Delta V}{3}\frac{1}{\sum_j\left(\sum_{(i,k)}\beta_{ijk}\left((x_i+u_i^x)\,z_k\right)\right)^2}$$

Enfin, en reportant dans la première égalité, on obtient

$$u_j^y = \frac{\Delta V}{3}\frac{\displaystyle\sum_{(i,k)}\beta_{ijk}\left((x_i+u_i^x)\,z_k\right)}{\displaystyle\sum_{j'}\left(\sum_{(i,k)}\beta_{ij'k}\left((x_i+u_i^x)\,z_k\right)\right)^2}$$

En utilisant la notation ${}^*V_j^{XZ} = \sum_{(i,k)}\beta_{ijk}\ ((x_i+u_i^x)\ z_k)$, on peut alors mettre le résultat sous la forme simplifiée

$$\mathbf{u}^y = \frac{\Delta V}{3}\frac{{}^*\mathbf{V}^{XZ}}{\|{}^*\mathbf{V}^{XZ}\|^2}$$

Finalement, on peut, de même, exprimer le système à résoudre pour le dernier déplacement selon z.

$$\begin{cases} \forall k,\ 2u_k^z \\ +\lambda \sum_{(i,j)} \beta_{ijk} \left((x_i + u_i^x) \left(y_i + u_j^y \right) \right) = 0 \\ \\ \sum_{(i,j)} \beta_{ijk} \left((x_i + u_i^x) \left(y_i + u_j^y \right) z_k \right) = \dfrac{\Delta V}{3} \end{cases}$$

La méthode de résolution étant strictement identique, le résultat peut donc s'exprimer sans surprises par

$$u_k^z = \frac{\Delta V}{3} \frac{\sum_{(i,j)} \beta_{ijk} \left((x_i + u_i^x) \left(y_j + u_j^y \right) \right)}{\sum_{k'} \left(\sum_{(i,j)} \beta_{ijk'} \left((x_i + u_i^x) \left(y_j + u_j^y \right) \right) \right)^2}$$

ou bien, de façon simplifiée par

$$\mathbf{u}^z = \frac{\Delta V}{3} \frac{{}^{**}\mathbf{V}^{XY}}{\|{}^{**}\mathbf{V}^{XY}\|^2} \ ,$$

avec ${}^{**}V_k^{XZ} = \sum_{(i,j)} \beta_{ijk} \left[(x_i + u_i^x) \left(y_j + u_j^y \right) \right]$.

Annexe C

Correction de volume pondéré sur les sommets dans le cas exact

On recherche cette fois à résoudre la minimisation pondérée donnée dans la section 4.5. La minimisation sous contrainte s'exprime cette fois par

$$\begin{cases} \min & \displaystyle\sum_i \left[\frac{\|\mathbf{u}_i\|^2}{\gamma_i}\right] \\ \text{contraint à} & V(\mathbf{x}+\mathbf{u}) = V^0 \ , \end{cases} \tag{C.1}$$

On suppose toujours qu'un tiers des corrections est portée par chacun des trois axes.

On obtient donc, par la méthode des multiplicateurs de Lagrange, les trois fonctions d'énergie à résoudre dans l'ordre

$$\begin{cases} E_1 = \displaystyle\sum_{i=0}^{N_s} \left[\frac{(u_i^x)^2}{\gamma_i}\right] + \lambda \left[\sum_{(i,j,k)} \beta_{ijk}\ u_i^x y_j z_k - \frac{\Delta V}{3}\right] \\ E_2 = \displaystyle\sum_{i=0}^{N_s} \left[\frac{(u_i^y)^2}{\gamma_i}\right] + \lambda \left[\sum_{(i,j,k)} \beta_{ijk}\ (x_i + u_i^x) u_j^y z_k - \frac{\Delta V}{3}\right] \\ E_3 = \displaystyle\sum_{i=0}^{N_s} \left[\frac{(u_i^z)^2}{\gamma_i}\right] + \lambda \left[\sum_{(i,j,k)} \beta_{ijk}\ (x_i + u_i^x)(y_j + u_j^y) u_k^z - \frac{\Delta V}{3}\right] \end{cases}$$

Comme pour les méthodes précédentes, la résolution se réalise de façon identique en écrivant l'annulation des dérivées partielles. Ainsi, pour le cas de l'axe x, on peut écrire

$$\begin{cases} \forall i,\ 2\,\dfrac{u_i^x}{\gamma_i} + \lambda \displaystyle\sum_{(j,k)} \beta_{ijk}\ y_j z_k = 0 \\ \displaystyle\sum_{(i,j,k)} \beta_{ijk}\ u_i^x y_j z_k = \frac{\Delta V}{3} \end{cases}$$

On peut alors isoler u_i^x dans la première égalité et l'injecter dans la seconde pour en tirer

$$\lambda = -\frac{2\,\Delta V}{3} \frac{1}{\sum_i \gamma_i \left(\sum_{(j,k)} \beta_{ijk}\ y_j z_k\right)^2}$$

On en déduit alors la composante du déplacement

$$u_i^x = \frac{\Delta V}{3} \frac{\gamma_i \sum_{(j,k)} \beta_{ijk}\, y_j z_k}{\sum_{i'} \left[\gamma_{i'} \left(\sum_{(j,k)} \beta_{i'jk}\, y_j z_k \right)^2 \right]}$$

De façon triviale, on peut désormais exprimer les deux autres composantes

$$u_j^y = \frac{\Delta V}{3} \frac{\gamma_j \sum_{(i,k)} \beta_{ijk}\ (x_i + u_i^x)\, z_k}{\sum_{j'} \left[\gamma_{j'} \left(\sum_{(i,k)} \beta_{ij'k}\ (x_i + u_i^x)\, z_k \right)^2 \right]}$$

$$u_k^z = \frac{\Delta V}{3} \frac{\gamma_k \sum_{(i,j)} \beta_{ijk}\ (x_i + u_i^x) \left(y_j + u_j^y \right)}{\sum_{k'} \left[\gamma_{k'} \left(\sum_{(i,j)} \beta_{ijk'}\ (x_i + u_i^x) \left(y_j + u_j^y \right) \right)^2 \right]}$$

En utilisant désormais les notations

$$\begin{cases} W_i^{YZ} = \gamma_i \sum_{(j,k)} \beta_{ijk}\, y_j z_k \\ {}^{*}W_j^{XZ} = \gamma_j \sum_{(i,k)} \beta_{ijk}\, (x_i + u_i^x) z_k \\ {}^{**}W_k^{XY} = \gamma_k \sum_{(i,j)} \beta_{ijk}\, (x_i + u_i^x)(y_j + u_j^y) \end{cases}$$

On peut évidement exprimer les résultats sous la forme vectorielle

$$\begin{cases} \mathbf{u}^x = \dfrac{\Delta V}{3} \dfrac{\mathbf{W}^{YZ}}{\langle \mathbf{W}^{YZ}, \mathbf{V}^{YZ} \rangle} \\ \mathbf{u}^y = \dfrac{\Delta V}{3} \dfrac{{}^{*}\mathbf{W}^{XZ}}{\langle {}^{*}\mathbf{W}^{XZ}, {}^{*}\mathbf{V}^{XZ} \rangle} \\ \mathbf{u}^z = \dfrac{\Delta V}{3} \dfrac{{}^{**}\mathbf{W}^{XY}}{\langle {}^{**}\mathbf{W}^{XY}, {}^{**}\mathbf{V}^{XY} \rangle} \end{cases}$$

Annexe D

Contrainte de positions fixes

On cherche cette fois à résoudre la minimisation pour la contrainte de volume dans les trois directions avec la contrainte supplémentaire de positions fixes. Cette contrainte est introduite en section 4.6.

D.1 Minimisation à résoudre

On veut cette fois garder un ensemble de positions fixes. Les sommets fixes sont pointés par les index contenus dans l'ensemble $\mathcal{F}$.

On considère toujours trois contraintes séparées en x, y et z. La minimisation en x peut alors s'exprimer par

$$\begin{cases} \text{min} & \displaystyle\sum_i \frac{(u_i^x)^2}{\gamma_i} \\ \text{contraint à} & \displaystyle\sum_{(i,j,k)} \beta_{ijk}\, u_i^x y_j z_k = \frac{\Delta V}{3} \\ \text{contraint à} & \displaystyle\sum_{i \notin \mathcal{F}} (u_i^x)^2 = 0 \end{cases}$$

Le Lagrangien s'exprime alors par

$$E_1(\mathbf{u}^x, \lambda, \mu) = \sum_i \frac{(u_i^x)^2}{\gamma_i} + \lambda \left[\sum_{(i,j,k)} \beta_{ijk}\, u_i^x y_j z_k - \frac{\Delta V}{3} \right] + \mu \sum_{i \notin \mathcal{F}} (u_i^x)^2$$

Le système à résoudre est donc donné cette fois par l'annulation des dérivées partielles par rapport aux u_i^x, λ et μ.

$$\begin{cases} \forall i\,, \;\; 2\,\dfrac{u_i^x}{\gamma_i}\left(1 + \mu\,\delta_E^i\right) + \lambda \displaystyle\sum_{(j,k)} \beta_{ijk}\, y_j z_k = 0 \\ \displaystyle\sum_{(i,j,k)} \beta_{ijk}\, u_i^x y_j z_k = \frac{\Delta V}{3} \\ \forall i \notin \mathcal{F}, \;\; u_i^x = 0 \end{cases}$$

où $\delta_E^i = 1$ si i est dans l'ensemble $\mathcal{F}$ et vaut zéro sinon.

D.2 Résolution

On peut alors écrire la somme de la seconde égalité comme

$$\sum_{(i,j,k)} \beta_{ijk}\, u_i^x y_j z_k = \underbrace{\sum_{i\notin\mathcal{F}} \sum_{(j,k)} \beta_{ijk}\, u_i^x y_j z_k}_{=0} + \sum_{i\in\mathcal{F}} \sum_{(j,k)} \beta_{ijk}\, u_i^x y_j z_k$$

On en déduit donc que la contrainte sur le volume est donnée par

$$\sum_{i\in\mathcal{F}} \sum_{(j,k)} \beta_{ijk}\, u_i^x y_j z_k = \frac{\Delta V}{3} \ . \tag{D.1}$$

D'un autre côté, on peut toujours exprimer u_i^x d'après la première égalité par

$$u_i^x = -\frac{\lambda}{2(1+\mu\,\delta_E^i)} \gamma_i \sum_{(j,k)} \beta_{ijk}\, y_j z_k \ .$$

En injectant cette égalité dans l'eq. D.1, on obtient alors

$$-\frac{\lambda}{2} \sum_{i\in F} \left[\frac{\gamma_i}{1+\mu\,\delta_E^i} \left(\sum_{(j,k)} \beta_{ijk}\, y_j z_k \right)^2 \right] = \frac{\Delta V}{3} \ .$$

Or δ_E^i s'annule pour chaque $i \in \mathcal{F}$, on peut donc exprimer directement

$$\lambda = -\frac{2\,\Delta V}{3} \frac{1}{\sum_{i\in\mathcal{F}} \gamma_i \left(\sum_{(j,k)} \beta_{ijk}\, y_j z_k \right)^2}$$

Et donc la correction s'exprime directement par

$$u_i^x = \begin{cases} \dfrac{\Delta V}{3} \dfrac{\gamma_i \displaystyle\sum_{(j,k)} \beta_{ijk}\, y_j z_k}{\displaystyle\sum_{i'\in\mathcal{F}} \gamma_{i'} \left(\sum_{(j,k)} \beta_{i'jk}\, y_j z_k \right)^2} & \text{pour } i \in \mathcal{F} \\ 0 & \text{pour } i \notin \mathcal{F} \end{cases}$$

On en déduit, de façon identique le calcul pour y et z

$$u_j^y = \begin{cases} \dfrac{\Delta V}{3} \dfrac{\gamma_j \displaystyle\sum_{(i,k)} \beta_{ijk}\, (x_i + \delta_i^x) z_k}{\displaystyle\sum_{j'\in\mathcal{F}} \gamma_{j'} \left(\sum_{(i,k)} \beta_{ij'k}\, (x_i + \delta_i^x) z_k \right)^2} \\ \text{pour } j \in \mathcal{F} \\ 0 \\ \text{pour } j \notin \mathcal{F} \end{cases}$$

$$u_k^z = \begin{cases} \dfrac{\Delta V}{3} \dfrac{\gamma_k \displaystyle\sum_{(i,j)} \beta_{ijk}\ (x_i + \delta_i^x)(y_j + \delta_j^y)}{\displaystyle\sum_{k' \in \mathcal{F}} \gamma_{k'} \left(\sum_{(i,j)} \beta_{ijk'}\ (x_i + \delta_i^x)(y_j + \delta_j^y) \right)^2} \\ \text{pour } k \in \mathcal{F} \\ \\ 0 \\ \text{pour } k \notin \mathcal{F} \end{cases}$$

Finalement, en appelant le vecteur $\mathbf{a}_{|\mathcal{F}}$ tel que $a_{i_{|\mathcal{F}}} = a_i\, \delta_{\mathcal{F}}^i$, on peut évidemment écrire le résultat sous la forme vectorielle condensée en utilisant les notations précédentes.

$$\begin{cases} \mathbf{u}^x = \dfrac{\Delta V}{3} \dfrac{\mathbf{W}_{|\mathcal{F}}^{YZ}}{\left\langle \mathbf{W}_{|\mathcal{F}}^{YZ}, \mathbf{V}_{|\mathcal{F}}^{YZ} \right\rangle} \\ \\ \mathbf{u}^y = \dfrac{\Delta V}{3} \dfrac{{}^*\mathbf{W}_{|\mathcal{F}}^{XZ}}{\left\langle {}^*\mathbf{W}_{|\mathcal{F}}^{XZ}, {}^*\mathbf{V}_{|\mathcal{F}}^{XZ} \right\rangle} \\ \\ \mathbf{u}^z = \dfrac{\Delta V}{3} \dfrac{{}^{**}\mathbf{W}_{|\mathcal{F}}^{XY}}{\left\langle {}^{**}\mathbf{W}_{|\mathcal{F}}^{XY}, {}^{**}\mathbf{V}_{|\mathcal{F}}^{XY} \right\rangle} \end{cases}$$

Annexe E

Correction de volume dans la direction normale dans le cas linéarisé

On cherche à résoudre la minimisation suivante (approximation linéaire)

$$\begin{cases} \min & \displaystyle\sum_i \frac{\rho_i^2}{\gamma_i} \\ \\ \text{contraint à} & V(\rho\, n^x, y, z) + V(x, \rho\, n^y, z) + V(x, y, \rho\, n^z) = \Delta V \end{cases} \tag{E.1}$$

L'expression du Lagrangien est donc donnée par

$$E = \sum_i \frac{\rho_i^2}{\gamma_i} + \lambda \left[\sum_{(i,j,k)} \beta_{ijk} \Big(\rho_i n_i^x y_j z_k + x_i \rho_j n_j^y z_k + x_i y_j \rho_k n_k^z \Big) - \Delta V \right]$$

La solution s'exprime donc par le système

$$\begin{cases} \forall m, \quad \dfrac{\partial E}{\partial \rho_m} = 0 \\ \\ \dfrac{\partial E}{\partial \lambda} = 0 \end{cases}$$

soit

$$\begin{cases} 2\dfrac{\rho_m}{\gamma_m} + \lambda \left[n_m^x \displaystyle\sum_{(j,k)} \beta_{mjk}\, y_j z_k + n_m^y \sum_{(i,k)} \beta_{imk}\, x_i z_k + n_m^z \sum_{(i,j)} \beta_{ijm}\, x_i y_j \right] = 0 \\ \\ \displaystyle\sum_{(i,j,k)} \beta_{ijk} \Big(\rho_i n_i^x y_i z_k + x_i \rho_j n_j^y z_k + x_i y_j \rho_k n_k^z \Big) = \Delta V \end{cases}$$

Pour simplifier les expressions, on pose

$$K_m = n_m^x \sum_{(j,k)} \beta_{mjk}\, y_j z_k + n_m^y \sum_{(i,k)} \beta_{imk}\, x_i z_k + n_m^z \sum_{(i,j)} \beta_{ijm}\, x_i y_j \ .$$

On tire donc de la première égalité

$$\rho_m = -\frac{\lambda}{2} \gamma_m\, K_m \ .$$

En injectant cette expression dans la seconde égalité, on obtient donc

$$-\frac{\lambda}{2}\left[\sum_{(i,j,k)} \beta_{ijk}\Big(\gamma_i K_i n_i^x y_j z_k + x_i \gamma_j K_j n_j^y z_k + x_i y_j \gamma_k K_k n_k^z\Big)\right] = \Delta V \ ,$$

soit

$$\lambda = -\frac{2\Delta V}{\left(\sum_i \gamma_i n_i^x K_i \sum_{(j,k)} \beta_{ijk} y_j z_k\right) + \left(\sum_j \gamma_j n_j^y K_j \sum_{(i,k)} \beta_{ijk} x_i z_k\right) + \left(\sum_k \gamma_k n_k^z K_k \sum_{(i,j)} \beta_{ijk} x_i y_j\right)}$$

On obtient donc, en rééxprimant la première égalité du système la valeur de ρ_m

$$\rho_m = \Delta V \frac{\gamma_m \left(n_m^x \sum_{(j,k)} \beta_{mjk}\, y_i z_k + n_m^y \sum_{(i,k)} \beta_{imk}\, x_i z_k + n_m^z \sum_{(i,j)} \beta_{ijm}\, x_i y_j\right)}{\left(\sum_i \gamma_i n_i^x K_i \sum_{(j,k)} \beta_{ijk} y_j z_k\right) + \left(\sum_j \gamma_j n_j^y K_j \sum_{(i,k)} \beta_{ijk} x_i z_k\right) + \left(\sum_k \gamma_k n_k^z K_k \sum_{(i,j)} \beta_{ijk} x_i y_j\right)}$$

On peut utiliser les expressions

$$\begin{cases} V_i^{YZ} = \sum_{(j,k)} \beta_{ijk}\ y_j z_k \\ V_j^{XZ} = \sum_{(i,k)} \beta_{ijk}\ x_j z_k \\ V_k^{XY} = \sum_{(i,j)} \beta_{ijk}\ x_i y_j \end{cases}$$

On a alors

$$K_m = n_m^x\, V_m^{YZ} + n_m^y\, V_m^{XZ} + n_m^z\, V_m^{XY} \ ,$$

et donc

$$\rho_m = \Delta V \frac{\gamma_m K_m}{\sum_i \gamma_i n_i^x K_i V_i^{YZ} + \sum_j \gamma_j n_j^y K_j V_j^{XZ} + \sum_k \gamma_k n_k^z K_k V_k^{YZ}}$$

En rassemblant le dénominateur, on obtient alors

$$\rho_m = \Delta V \frac{\gamma_m K_m}{\sum_i \gamma_i K_i \left(n_i^x V_i^{YZ} + n_i^y V_i^{XZ} + n_i^z V_i^{YZ}\right)}$$

Enfin, en exprimant le vecteur de $\mathbb{R}^3$ $\mathbf{v}_i = (V_i^{YZ}, V_i^{XZ}, V_i^{YZ})$ et en remarquant que $K_m =< \mathbf{n}_m, \mathbf{v}_m >$, on peut alors exprimer l'expression grâce aux produits scalaires

$$\rho_m = \Delta V \frac{\gamma_m < \mathbf{n}_m, \mathbf{v}_m >}{\sum_i \gamma_i < \mathbf{n}_i, \mathbf{v_i} >^{\mathbf{2}}}$$

Annexe F

Correction de volume dans le cas des coordonnées laplaciennes

On cherche à résoudre la minimisation suivante

$$\begin{cases} \min & \displaystyle\sum_i \frac{(\rho_i)^2}{\gamma_i} + \kappa \sum_i \frac{1}{\gamma_i} \, \|\mathcal{L}(\rho\,\mathbf{n}_i)\|^2 \\ \\ \text{contraint à} & V(\mathbf{x} + \rho\,\mathbf{n}) - V^0 = 0 \ , \end{cases}$$

qui se met donc sous la forme de la fonctionnelle d'energie

$$E = \sum_i \frac{(\rho_i)^2}{\gamma_i} + \kappa \sum_i \frac{1}{\gamma_i} \, \|\mathcal{L}(\rho\,\mathbf{n}_i)\|^2 + \lambda \left[V(\mathbf{x} + \rho\,\mathbf{n}) - V^0 \right]$$

On peut développer cette expression dans le cas où l'on linéarise l'expression du volume par

$$\begin{aligned} E = \sum_i \frac{(\rho_i)^2}{\gamma_i} \quad & + \kappa \sum_i \sum_{j \in \mathcal{V}_i} \frac{1}{\mathcal{N}_i \, \gamma_i} \left[\left(\rho_i n_i^x - \rho_j n_j^x\right)^2 + \left(\rho_i n_i^y - \rho_j n_j^y\right)^2 + \left(\rho_i n_i^z - \rho_j n_j^z\right)^2 \right] \\ & + \lambda \left[\sum_{(i,j,k)} \beta_{ijk} \left(\rho_i n_i^x y_j z_k + x^i \rho_j n_j^y z_k + x^i y_j \rho_k n_k^z \right) - \Delta V \right] \end{aligned}$$

Afin de résoudre la minimisation, il est nécessaire d'annuler les dérivées partielles par rapport aux ρ_i.

$$\forall m \in [\![0, N_s]\!] \ , \quad \frac{\partial E}{\partial \rho_m} = 0$$

$$\begin{aligned} \Rightarrow 2\frac{\rho_m}{\gamma_m} \quad & + \kappa \sum_{j \in \mathcal{V}_m} \frac{2}{\mathcal{N}_m \, \gamma_m} \Big(\left[\rho_m \, (n_m^x)^2 - n_m^x \rho_j n_j^x\right] + \left[\rho_m \, (n_m^y)^2 - n_m^y \rho_j n_j^y\right] + \left[\rho_m \, (n_m^z)^2 - n_m^z \rho_j n_j^z\right] \Big) \\ & + \sum_i \frac{2}{\mathcal{N}_i \gamma_i} \delta_{\mathcal{V}_i}^m \Big(\left[\rho_m (n_m^x)^2 - \rho_i n_i^x n_m^x\right] + \left[\rho_m (n_m^y)^2 - \rho_i n_i^y n_m^y\right] + \left[\rho_m (n_m^z)^2 - \rho_i n_i^z n_m^z\right] \Big) \\ & + \lambda \left(n_m^x \sum_{(j,k)} \beta_{mjk} \, y_j z_k + n_m^y \sum_{(i,k)} \beta_{imk} \, x_i z_k + n_m^z \sum_{(i,j)} \beta_{ijm} \, x_i y_j \right) = 0 \ , \end{aligned}$$

où l'on utilise la notation $\delta_{\mathcal{V}_m}^i$ valant un si $i \in \mathcal{V}_m$ et zéro sinon.

On remarquera qu'en posant à nouveau les notations

$$K_m = n_m^x \sum_{(j,k)} \beta_{mjk} \, y_j z_k + n_m^y \sum_{(i,k)} \beta_{imk} \, x_i z_k + n_m^z \sum_{(i,j)} \beta_{ijm} \, x_i y_j \ ,$$

en appelant

$$\mathbf{p}_m = \frac{1}{\mathcal{N}_m} \sum_{i \in \mathcal{V}_m} \rho_i \mathbf{n}_i \ , \text{ et } \tilde{\mathbf{p}}_m = \sum_i \frac{\delta^m_{\mathcal{V}_i}}{\mathcal{N}_i \, \gamma_i} \rho_i \mathbf{n}_i \ ,$$

on peut exprimer la relation précédente par

$$2\frac{\rho_m}{\gamma_m} + 2\kappa \left[\rho_m \|\mathbf{n}_m\|^2 \left(\frac{1}{\gamma_m} + \sum_i \frac{\delta^m_{\mathcal{V}_i}}{\mathcal{N}_i \gamma_i} \right) - \left\langle \mathbf{n}_m, \frac{\mathbf{p}_m}{\gamma_m} - \tilde{\mathbf{p}}_m \right\rangle \right] + \lambda K_m = 0$$

La relation donne cette fois lieu à un système en ρ_i et on ne peut donc pas exprimer directement ρ_m en fonction de λ.

On rassemble dans le vecteur $\boldsymbol{\rho}$ les ρ_i correspondant à chaque sommet. On peut alors écrire le système sous la forme matricielle

$$\mathrm{A}\, \boldsymbol{\rho} = -\lambda \, \mathbf{K} \ ,$$

où $\mathbf{K}$ correspond au vecteur des K_m.

On peut alors écrire les coefficients de la matrice A

$$\mathrm{A}_{ij} = \frac{2}{\gamma_i} \left(1 + \kappa \|\mathbf{n}_i\|^2\right) + 2\kappa \left[\sum_k \frac{\delta^i_k}{\mathcal{N}_k \gamma_k} \|\mathbf{n}_i\|^2 - \left\langle \mathbf{n}_i, \left(\frac{\delta^j_{\mathcal{V}_i}}{\mathcal{N}_i \, \gamma_i} + \frac{\delta^i_{\mathcal{V}_j}}{\mathcal{N}_j \, \gamma_j} \right) \mathbf{n}_j \right\rangle \right]$$

Il est évident que cette matrice est à diagonale dominante, en effet

$$|\mathrm{A}_{ii}| = \left| \frac{2}{\gamma_i} \left(1 + \kappa \|\mathbf{n}_i\|^2\right) + 2\kappa \sum_k \frac{\delta^i_k}{\mathcal{N}_k \gamma_k} \|\mathbf{n}_i\|^2 \right| > \left| \frac{2}{\gamma_i} \kappa \|\mathbf{n}_i\|^2 \right|$$

et l'on a d'un autre côté

$$\begin{aligned} \forall (i,j) \, , \ \ |\mathrm{A}_{ij}| \ &= \left| -\kappa \left\langle \mathbf{n}_i, \left(\frac{\delta^j_{\mathcal{V}_i}}{\mathcal{N}_i \, \gamma_i} + \frac{\delta^i_{\mathcal{V}_j}}{\mathcal{N}_j \, \gamma_j} \right) \mathbf{n}_j \right\rangle \right| \\ &< \left| \kappa \left\langle \mathbf{n}_i, \left(\frac{1}{\mathcal{N}_i \, \gamma_i} + \frac{1}{\mathcal{N}_j \, \gamma_j} \right) \mathbf{n}_j \right\rangle \right| \\ &< \frac{1}{\mathcal{N}_i} \left| \kappa \frac{2}{\gamma_i} \, \kappa \|\mathbf{n}_i\|^2 \right| \ , \end{aligned}$$

d'où

$$\forall (i,j) \, , \ \ |\mathrm{A}_{ii}| < |\mathrm{A}_{ij}| \ .$$

La matrice peut donc s'écrire sous la forme

$$\mathrm{A} = \mathrm{D} + \varepsilon \ ,$$

où D est une matrice diagonale et où ε est une matrice très creuse en ne considérant que le voisinage direct de chaque sommet. Cette matrice est donc évidemment inverssible.

La solution du système $N_s \times N_s$ est donc donné sous la forme

$$\rho = -\lambda \, (\mathrm{A})^{-1} \mathbf{K} \ .$$

Posons $\mathbf{q} = (\mathrm{A})^{-1}\mathbf{K}$. La seconde condition de résolution par les multiplicateurs de Lagrange est donnée par

$$\sum_{(i,j,k)} \beta_{ijk} \left(\rho_i n^x_i y_j z_k + x_i \rho_j n^y_j z_k + x_i y_j \rho_k n^z_k \right) = \Delta V$$

En injectant l'expression de ρ_i dans cette expression, on obtient alors de façon identique aux autres résolutions

$$\lambda = -\frac{\Delta V}{\sum_i q_i < \mathbf{n}_i, \mathbf{v}_i >} \ ,$$

en reposant de façon identique $\mathbf{v}_i = (V_i^{YZ}, V_i^{XZ}, V_i^{XY})$. On peut donc exprimer la variation à réaliser par

$$\forall i\,,\ \rho_i = \Delta V \frac{q_i}{\sum_j q_j \ < \mathbf{n}_j, \mathbf{v}_j >} \ .$$

Bibliographie

[ABE07] D. Attali, J-D. Boissonnat, and H. Edelsbrunner. *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, chapter Stability and Computation of Medial Axes : a State of the Art Report. Springer-Verlag, 2007.

[ACWK04] Alexis Angelidis, Marie-Paule Cani, Geoff Wyvill, and Scott King. Swirling-sweepers : Constant-volume modeling. *Pacific Graphics*, October 2004.

[AHU59] Kenneth J. Arrowa, Leonid Hurwicz, and Hirofumi Uzawa. Studies in linear and non-linear programming. *The American Economic Review*, 49(4) :751–754, September 1959.

[Ale02] Marc Alexa. Linear combination of transformations. *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 380–387, 2002.

[And01] Eike F Anderson. Real-time character animation for computer games. *Technical Report, National Centre for Computer Animation, Bournemouth University*, 2001.

[And05] Kasper A. Andersen. Spherical blend skinning on gpu. *Symposium on Interactive 3D Graphics*, pages 9–16, 2005.

[AWC04] Alexis Angelidis, Geoff Wyvill, and Marie-Paule Cani. Sweepers : Swept user-defined tools for modeling by deformation. *IEEE : Proceedings of Shape Modeling and Applications*, pages 63–73, June 2004.

[Bar84] Alan H. Barr. Global and local deformation of solid primitives. *SIGGRAPH '84 Proceedings of the 11th annual conference on Computer Graphics and Interactive Techniques*, 18 :21–30, 1984.

[BC00] David Bourguignon and Marie-Paule Cani. Controlling anisotropy in mass-spring systems. *Eurographics Workshop on Computer Animation and Simulation*, pages 113–123, August 2000.

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

[Blo02] Jules Bloomenthal. Medial-based vertex deformation. *Symposium on Computer Animation*, 2002.

[BS91] Jules Bloomenthal and Ken Shoemake. Convolution Surfaces. *SIGGRAPH'91 : Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 251–256, 1991.

[CB02] Erika Chuang and Chris Bregler. Performance driven facial animation using blendshape interpolation. *Technical report CS-TR-2002-02, Stanford University*, 2002.

[CD97] Marie-Paule Cani and Mathieu Desbrun. Animation of Deformable Models Using Implicit Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3 :39–50, 1997.

[CH01] Gordon Collins and Adrian Hilton. Models for character animation. *Software Focus*, 2(2) :44–51, 2001.

[Che92] David Tzu-Wei Chen. *Pump It Up : computer animation of a biomechanically based model of muscle using the finite element method*. PhD thesis, Massachusetts Institute of Technology. Dept. of Architecture, Massachusetts, USA, 1992.

[CHP89] John E. Chadwick, David R. Haumann, and Richard E. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3) :243–252, July 1989.

[Chu05] Erika Chuang. Mood Swings : Expressive Speech Animation. *ACM Transactions on Graphics*, 24(2) :331–347, April 2005.

[CN00] Clint Chua and Ulrich Neumann. Hardware-accelerated free-form deformation. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 33–39, 2000.

[Coq90] Sabine Coquillart. Extended free-form deformation : a sculpturing tool for 3d geometric modeling. *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer Graphics and Interactive Techniques*, pages 187–196, 1990.

[CR94] Yu-Kuang Chang and Alyn P. Rockwood. A generalized de casteljau approach to 3D free-form deformation. *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer Graphics and Interactive Techniques*, pages 257–260, 1994.

[CS06] Daniel Cohen-Or and Olga Sorkine. Encoding meshes in differential coordinates. *Spring Conference on Computer Graphics*, 2006.

[DC95] Mathieu Desbrun and Marie-Paule Cani. Animating soft substances with implicit surfaces. *SIGGRAPH*, 29 :287–290, August 1995.

[DKB+99] C. C. Van Donkelaar, I. J. G. Kretzers, P. H. M. Bovendeerd, L. M. A. Lataster, K. Nicolay, J. D. Janseen, and M. R. Drost. Diffusion tensor imaging in biomechanical studies of skeletal muscle function. *Journal of Anatomy*, 194 :79–88, January 1999.

[Dul04] Penka Dulgerova. Biophysical modelling of viscoelastic deformations of cattle skin in vivo. *Trakia Journal of Sciences*, 2(4) :1–3, 2004.

[EFFM02] Douglas Enright, Ronald Fedkiw, Joel Ferziger, and Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1) :83 – 116, November 2002.

[EMBL99] Laura E. Edsberg, Robert E. Mates, Robert E. Baier, and Mark Lauren. Mechanical characteristics of human skin subjected to static versus cyclic normal pressures. *Journal of Rehabilitation Research and Development*, 36(2) :133–141, April 1999.

[FF01] Nick Foster and Ronald Fedkiw. Practical animation of liquids. *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, 2001.

[FHW98] Jieqing Feng, Pheng-Ann Heng, and Tien-Tsin Wong. Accurate B-Spline free-form deformation of polygonal objects. *Journal of Graphics Tools*, 3(3) :11–27, March 1998.

[FLM03] Mark Foskey, Ming C. Lin, and Dinesh Manocha. Efficient computation of a simplified medial axis. *ACM Symposium on Solid Modeling and Applications*, 2003.

[FMP96] Jieqing Feng, Lizhuang Ma, and Qunsheng Peng. A new free-form deformation through the control of parametric surfaces. *Computers and Graphics*, 20(4) :531–539, July 1996.

[FPT97] Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. Dynamic free-form deformations for animation synthesis. *IEE Transactions on Visualization and Computer Graphics*, 3(3) :201–214, July 1997.

[FTS06] Wolfram Von Funck, Holger Theisel, and Hans-Peter Seidel. Vector field based shape deformations. *ACM Transactions on Graphics*, 25(3) :1118–1125, July 2006.

[GCP97] Carlos Gonzalez-Ochoa, Scott Mc Cammon, and Jörg Peters. Computing moments of piecewise polynomial surfaces. *Technical Report TR-97-009, Purdue University, Indiana USA*, January 1997.

[Gol91] Eliot Golfinger. *Human Anatomy for Artists : The Elements of Form*. Oxford University Press, 1991.

[GP89] Josef Griessmair and Werner Purgathofer. Deformation of solids with trivariate B-Splines. *Proceedings for Eurographics*, pages 137–148, September 1989.

[Hen01] F. M. Hendriks. Mechnical behaviour of human skin in vivo- A literature review. 2001.

[HHK92] William M. Hsu, John F. Hughes, and Henry Kaufman. Direct manipulation of free-form deformations. *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 177–184, July 1992.

[HML99] Gentaro Hirota, Renee Maheshwari, and Ming C. Lin. Fast volume-preserving free form deformation using multi-level optimization. *SMA '99 : Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 234–245, 1999.

[HSL+06] 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*, 25(3) :1126–1134, July 2006.

[HZTS01] Shi-Min Hu, Hui Zhang, Chiew-Lan Tai, and Jia-Guang Sun. Direct manipulation of ffd : Efficient explicit solutions and decomposible multiple point constraints. *Visual Computer*, 17(6) :370–379, July 2001.

[JT05] Doug L. James and Christopher D. Twigg. Skinning mesh animations. *ACM SIGGRAPH'05 : Transactions on Graphics*, 24(3) :399–407, August 2005.

[KCS06] Ladislav Kavan, Steven Collins, and Carol O' Sullivan. Dual quaternions for rigid transformation blending. *Technical Report, Trinity College, Dublin*, 2006.

[KCZS07] Ladislav Kavan, Steven Collins, Jiří Zára, and Carol O' Sullivan. Skinning with dual quaternions. *SIGGRAPH '07 : Symposium on Interactive 3D Graphics and Games*, April 2007.

[KJP02] Paul G. Kry, Doug L. James, and Dinesh K. Pai. Eigenskin : Real time large deformation character skinning in hardware. *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 153–159, 2002.

[KZ05] Ladislav Kavan and Jiří Zára. Spherical blend skinning : a real-time deformation of articulated models. *SI3D '05 : Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 9–16, 2005.

[Lar04] Caroline Larboulette. *Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages.* PhD thesis, INSA Rennes-EVASION INRIA, 2004.

[LCF00] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation : A unified approach to shape interpolation and skeleton-driven deformation. *SIGGRAPH'00, ACM Computer Graphics*, pages 85–94, Jully 2000.

[LCOGL06] Yaron Lipman, Daniel Cohen-Or, Ran Gal, and David Levin. Volume and shape preservation via moving frame manipulation. *Technical report, Tel-Aviv University*, 2006.

[LPJ01] Bent D. Larsen, Kim S. Petersen, and Bjarke Jakobsen. Deformable skinning on bones. *Technical Report University of Denmark*, June 2001.

[LSC+04] Yaron Lipman, Olga Sorkine, Daniel Cohen-Or, David Levin, Christian Rössl, and Hans-Peter Seidel. Differential coordinates for interactive mesh editing. *Shape Modeling International*, 2004.

[LSLC05] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or. Linear rotation-invariant coordinates for meshes. *Proceedings of ACM SIGGRAPH 2005*, pages 479–487, 2005.

[Mar98] Arne Marthinsen. Interpolation in lie groups and homogeneous spaces. *Technical Report n. 139, Department of Mathematics, Arizona State University, Tempe, Arizona.*, May 1998.

[MDSB02] Mark Meyer, Mathieur Desbrun, Peter Schröder, and Alan H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Proceedings of VisMath*, 2002.

[MG03] Alex Mohr and Michael Gleicher. Building efficient, accurate character skins from examples. *SIGGRAPH'03 :*, pages 562–568, 2003.

[MYF06] Hiroshi Masuda, YasuhiroYoshioka, and Yoshiyuki Furukawa. Preserving form-features in interactive mesh deformation. *Lecture Notes in Computer Science 4077 (Geometric Modeling and Processing)*, pages 207–220, 2006.

[Ned98] Luciana P. Nedel. *Anatomic Modeling of Human Bodies Using Physically-Based Muscle Simulation.* PhD thesis, École Polytechnique Fédérale de Lausanne, Suisse, 1998.

[NMK+06] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4) :809–836, December 2006.

[NSACO05] Andrew Nealen, Olga Sorkine, Marc Alexa, and Daniel Cohen-Or. A sketch-based interface for detail-preserving mesh editing. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 24(3) :1142–1147, 2005.

[OF03] Stanley Osher and Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

[OMO+03] C.W.J. Oomens, M. Maenhout, C.H. van Oijen, M.R. Drost, and F.P. Baaijens. Finite element modelling of contracting skeletal muscle. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 358 :1453–60, September 2003.

[OS88] Stanley Osher and James A. Sethian. Fronts propagating with curvature dependent speed : Algorithms based on Hamilton-Jacobi. *Journal of Computational Physics*, 79 :12–49, 1988.

[PH02] Michael Putz and Klaus Hufnagl. Character animation for real-time applications. *Central European Seminar on Computer Graphics*, 2002.

[PN97] Jörg Peters and Ahmad Nasri. Computing volumes of solids enclosed by recursive subdivision surfaces. *Computer Graphics Forum*, 16(3) :89–94, 1997.

[QSS00] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics.* Springer, 2000.

[RFDC05] Lionel Reveret, Laurent Favreau, Christine Depraz, and Marie-Paule Cani. Morphable model of quadrupeds skeletons for animating 3D animals. *ACM SIGGRAPH*, pages 29–31, July 2005.

[Rou04] Benjamin Rouveyrol. Animation skinnée par sémantique d'articulation. *Virtual Actors*, 2004.

[RSB96] Ari Rappoport, Alla Sheffer, and Michel Bercovier. Volume-preserving free-form solids. *IEEE Transactions on Visualization and Computer Graphics*, 2(1) :19–27, 1996.

[Sau05] Basile Sauvage. *Déformation de courbes et surfaces multirésolution sous contraintes.* PhD thesis, Institut National Polytechnique de Grenoble, Laboratoire Jean Kuntzmann, 2005.

[SCL+04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel. Laplacian surface editing. *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 179–188, 2004.

[SHB07] B. Sauvage, S. Hahmann, and G-P. Bonneau. Volume preservation of multiresolution meshes. *Computer Graphics Forum, Eurographics*, 26(3), 2007.

[SHBE06] B. Sauvage, S. Hahmann, G-P. Bonneau, and G. Elber. Detail preserving deformation of B-spline with volume constraint. Technical report, Research Report, 2006.

[Sho85] Ken Shoemake. Animating rotation with quaternion curves. *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 19(3) :245–254, July 1985.

[SK00] Karan Singh and Evangelos Kokkevis. Skinning characters using surface oriented free-form deformations. *Graphics Interface*, pages 35–42, 2000.

[Sor06] Olga Sorkine. Differential representations for mesh processing. *Computer Graphics Forum*, 25(4), December 2006.

[SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometry. *SIGGRAPH '86 : Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 20(4) :151–160, 1986.

[SSFD02] Frederick H. Sikver, Gurinder P. Seehra, Joseph W. Freeman, and Dale DeVore. Viscoelastic Properties of Young and Old Human Dermis : A Proposed Molecular Mechanism for Elastic Energy Storage in Collagen and Elastin. *Journal of Applied Polymer Science*, 86(8) :1978–1985, November 2002.

[TCET05] H.V Tran, F. Charleux, A. Ehrlacher, and M.C Ho Ba Tho. Propriétés mécaniques multi-couches de la peau humaine *in vivo*. *7ème Colloque National en calcul des structures*, May 2005.

[TQ94] Demetri Terzopoulos and Hong Qin. Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 13(2) :103–136, April 1994.

[WKT97] Yin Wu, Prem Kalra, and Nadia M. Thalmann. Physically-based wrinkle simulation and skin rendering. *Proc. Eurographics Workshop on Computer Simulation and Animation*, September 1997.

[WP02] Xiaohuan C. Wang and Cary Phillips. Multi-weight enveloping : least-squares approximation techniques for skin animation. *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 129–138, 2002.

[XZB06] Dong Xu, Hongxin Zhang, and Hujun Bao. Non-uniform differential mesh deformation. *Lecture Notes in Computer Science (Special Issue for Computer Graphics International)*, 2006.

[XZWB05] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao. Poisson shape interpolation. *ACM Symposium on Solid and Physical Modeling*, pages 267–274, 2005.

[YZ05] X. S. Yang and J. J. Zhang. Realistic skeleton driven skin deformation. *Lecture Notes in Computer Science (TSCG 2005), Springer-Verlag*, May 2005.

[YZX+04] 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*, pages 644–651, 2004.

[ZHS+05] 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 SIGGRAPH*, 24(3) :496–503, July 2005.

[ZL05] Xianlian Zhou and Jia LuIn. Nurbs-based galerkin method and application to skeletal muscle modeling. *In Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, pages 71–78, 2005.