UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathematique-Informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Damien ROHMER

Thèse dirigée par **Stefanie HAHMANN**
et codirigée par **Marie-Paule CANI**

préparée au sein **du Laboratoire Jean Kuntzmann (LJK),**
UMR CNRS 5224**.**
et de **l'école doctorale EDMSTII**
Mathématiques, Sciences et Technologies de l'Information, Informatique

Géométrie active pour l'animation et la modélisation

Thèse soutenue publiquement le **20 Septembre 2011**,
devant le jury composé de :

Valérie PERRIER
Professeur, Université de Grenoble, Présidente
Bruno LÉVY
Directeur de recherche, INRIA, Rapporteur
Eric GALIN
Professeur, Université de Lyon, Rapporteur
Loic BARTHE
Maitre de Conférence HDR, Université de Toulouse, Examinateur
Stefanie HAHMANN
Professeur, Université de Grenoble, INRIA, Directeur de thèse
Marie-Paule CANI
Professeur, Université de Grenoble, INRIA, Co-Directeur de thèse

RÉSUMÉ

Géométrie active pour l'animation et la modélisation

Engendrer en temps-réel des déformations visuellement réalistes d'objets 3D, comme par exemple le corps et les vêtements de personnages, est un point crucial en animation, que ce soit pour des applications au jeu vidéo ou à la production cinématographique.
Les méthodes de déformations géométriques actuelles rapides telles que le *skinning* ou l'animation physique à basse résolution ne capturent cependant pas certains comportements naturels essentiels. En particulier les déformations à volume constant du corps, le gonflement des muscles pour un personnage, ou la génération de plis sur ses vêtements dus au fait que leurs surfaces doivent rester développable.

Cette thèse présente une série de méthodes rendant les modèles géométriques *actifs*, c'est à dire capables de maintenir un certain nombre de contraintes intrinsèques de la surface portant sur le volume englobé ou sur le caractère développable de celle-ci. Nous étudions trois exemples :

- l'ajout de contraintes locales de volumes lors de l'animation d'une créature virtuelle par skinning.
- l'ajout de plis de vêtements modélisant une surface quasi-inextensible à partir d'une animation à basse résolution donnée en entrée.
- la génération d'une surface de type papier froissé, basée sur la préservation de l'isométrie vis-à-vis d'un patron planaire.

Dans tous ces modèles, notre approche est procédurale. Elle se base sur la déformation progressive et, potentiellement, le raffinement dynamique de la géométrie juste avant l'étape de rendu.

ABSTRACT

Active Geometry for Animation and Modeling

Generating visually realistic deformation of 3D objects in real-time for character body or garment is essential for the animation field, both for video games applications and movie production.
Current fast deformation methods such as *skinning*, or low resolution physically based simulations fail to capture some important natural behaviors. For instance constant volume deformation of a body, muscle bulging of a character, or wrinkling its garment which must be modeled by a developable surface.

This thesis presents several methods making a geometrical model *active*, that is to say able to maintain a certain number of intrinsic constraints of the surface linked to its bounded volume, or its developability. We develop three case studies

- The addition of local volume constraints during skinning animation of a virtual creature.
- The addition of wrinkles on garments to model stretch minimizing surface starting from a general low resolution animation provided as input.
- The generation of folded paper looking surface based on length preservation with respect to its planar pattern.

For every cases, we rely on a procedural approach based on progressive deformation, and eventually, on-the-fly refinement of the geometry just before the rendering stage.

REMERCIEMENTS

En tout premier lieu, j'adresse mes remerciements à mes deux encadrantes Stefanie Hahmann et Marie-Paule Cani qui ont toujours su me faire confiance et m'encourager du début jusqu'à la fin de cette thèse avec une grande sincérité.

Au cours de ces quatre dernières années, j'ai également eu la chance de côtoyer deux centres différents jour après jour, que ce soit sur le campus ou du côté INRIA.
A ce titre, j'adresse une pensée à la fois aux "anciens" thésards de la tour qui nous impressionnent toujours lorsque l'on débute : Claire, Irene, Carine, Laure, Ibrahim, Elie, Morgan, Souleymane, Azmi, Robin, Mathieu, Yann, fidèles compagnons de discutions du restaurant universitaire. Ainsi qu'à mes compatriotes qui ont partagé leurs débuts avec moi : Adrien, Roland, Madison, David. Avec un remerciement tout particulier à Emilie ainsi qu'à ma fidèle co-bureau de la tour Mélanie. Enfin, bien évidemment, je n'oublie pas Boris qui a toujours le bon mot pour rendre l'atmosphère plus sympathique.
Côté INRIA, je ne peux que remercier Seb pour sa bonne humeur et sa blagounette du matin ! J'adresse également un remerciement particulier à Estelle, Adrien, Romain avec qui j'ai passé ces dernières années. Et puis je n'oublie pas tous les autres doctorants, ingénieurs et stagiaires de l'équipe, avec entre autres, Adeline, Sahar, Marie, Dobrina, Berangère, Maxime, Jean-Remy, Simon, Cédric, Lucian, Benjamin, Guillaume, Francois, Antoine.
J'adresse également mes remerciements aux permanents qui ont toujours été enthousiastes et disponibles pour nous aider, que ce soit Georges-Pierre, Francois, Franck, Lionel, Jean-Claude, Olivier, Eric.

À ceci, je tiens enfin également remercier mes nouveaux collègues de CPE, avec, en particulier Jean-Marie et Catherine, qui m'ont fait toute confiance et donné ma chance afin de réaliser la jonction pour ma poursuite après thèse.

TABLE DES MATIÈRES

1 État de l'art **21**

1.1 Représentation par surfaces . . . 21
1.1.1 Modélisation explicite . . . 23
1.1.2 Modélisation implicite . . . 29
1.2 Déformation par modèles physiques . . . 37
1.2.1 Principe général . . . 37
1.2.2 Déformations physiques standard . . . 38
1.2.3 Cas particulier du modèle à masses-ressorts . . . 38
1.2.4 Animation physique de tissus . . . 41
1.3 Approches géométriques . . . 43
1.3.1 Déformations de l'espace . . . 44
1.3.2 Déformations par l'exemple . . . 62
1.3.3 Déformation basée surfaces . . . 65
1.4 Déformation sous contraintes . . . 70
1.4.1 Contraintes de volumes . . . 70
1.4.2 Méthodes de déformation à volume constant . . . 74
1.4.3 Contraintes de longueurs . . . 80

2 Animation de créatures virtuelles. **93**

2.1 Calcul automatique de poids de skinning . . . 95
2.1.1 Estimation rapide de poids à partir de la normale . . . 96
2.1.2 Calcul robuste des poids à partir d'information volumique . . . 97
2.2 Calcul du changement local de volume . . . 99
2.2.1 Estimation approximée rapide . . . 103
2.2.2 Calcul exact itératif . . . 105
2.3 Preservation locale de volume . . . 106
2.3.1 Préservation approximée . . . 107
2.3.2 Préservation exacte . . . 109

2.4 Controle de l'apparence finale . . . 111
2.4.1 Gestion automatique de l'apparence . . . 111
2.4.2 Controle utilisateur par courbes de profils . . . 113
2.5 Résultats : Comparaisons des différentes approches . . . 119
2.6 Évitement des auto-intersections . . . 128
2.7 Intégration pour une application de réalitée augmentée . . . 131
2.7.1 Articulation par transferts de déformations . . . 131
2.7.2 Déformations locales par squelette et maillage dynamique . . . 133
2.7.3 Préservation de volume itérative . . . 134
2.8 Extension aux contraintes d'aire constante . . . 136

3 Plis de vêtements 141
3.1 Génération statique de lignes de plis . . . 145
3.1.1 Champ de vecteurs issu des tenseurs d'élongation . . . 146
3.1.2 Tracé de courbes de plis . . . 148
3.2 Animation dynamique des courbes de plis . . . 150
3.2.1 Propagation des positions génératrices des courbes de plis . . . 151
3.2.2 Ajout et suppression de courbes . . . 152
3.2.3 Lissage temporel . . . 153
3.3 Déformeurs implicites . . . 154
3.3.1 Paramètres des plis . . . 154
3.3.2 Deformeurs géométriques implicites . . . 155
3.4 Resultats . . . 161
3.4.1 Mesure d'isométrie . . . 161
3.4.2 Réalisme visuel . . . 162
3.4.3 Application à des vêtements de personnages . . . 162

4 Papier froissé 171
4.1 Génération d'un bord 3D contrôlable . . . 173
4.2 Subdivision récursive d'une surface de papier froissé . . . 177
4.2.1 Génération de surface isométrique par diviser pour régner . . . 177
4.2.2 Calculs des lignes de plis . . . 178
4.2.3 Relaxation globale lorsque l'approche locale échoue . . . 183
4.3 Résultats de surfaces et quantification d'isométrie . . . 186
4.3.1 Construction de la surface . . . 186
4.3.2 Validation sur des surfaces sans plis . . . 186
4.3.3 Résultats sur des surfaces plissées . . . 188
4.3.4 Quantification de l'isométrie . . . 192
4.3.5 Extension au cas de surfaces non développables . . . 192
4.4 Limitations et travaux futurs . . . 194

5 Bilan et perspectives 197

A Calcul du volume 201
A.1 Cas continu . . . 201
A.2 Cas d'une surface polygonale . . . 202
A.3 Cas de polynomes triangulaires . . . 202
A.4 Volume associé à un carreau bilinéaires . . . 203

B Correction de volume **205**
B.1 Correction approchée suivant les trois axes 205
B.2 Correction approchée pondérée suivant la direction normale 206
B.3 Correction exacte pondérée en 3 étapes 207
B.4 Correction approchée de volume et d'aire suivant la direction normale 209

C Relation entre compression et profondeur du plis **211**
C.1 Relation entre compression et profondeur du plis 211
C.2 Pondération du noyaux de Cauchy à rayon fixé. 212

D Echantillonnage curviligne d'une cubique **213**

NOTATIONS UTILISÉES

Dans le reste de manuscrit, nous utiliserons les notations suivantes lorsque plus de précisions ne sont pas requises.

POINTS, VECTEURS ET MATRICES

Nous nous plaçons de manière générale dans l'espace 3D dans lequel nous associons un repère cartésien : $(O, \mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$. O étant l'origine du repère, $(\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$ étant trois vecteurs unitaires orthogonaux.
Ce repère défini des coordonnées absolues de l'espace qui seront notées (x, y, z). Un vecteur de l'espace 3D sera noté en gras : $\mathbf{a} = (a_x, a_y, a_z)$. Lorsque cela ne présente pas d'ambiguitée, nous confondrons la notion de points et vecteurs de l'espace. Une position spatiale quelconque sera généralement désignée par $\mathbf{p} = (x, y, z)$. Les vecteurs du plan 2D seront notés généralement par la lettre $\mathbf{q} = (x, y)$. Les matrices seront désignées par des lettres *Roman* comme par exemple

$$\mathrm{T} = \begin{pmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{pmatrix} .$$

NOTATION DIFFÉRENTIELLE

La dérivée d'une fonction f C^1 de plusieurs variables par rapport à une quantité scalaire u sera dénotée classiquement par $\partial f / \partial u$.

Dans le cas où f est une fonction scalaire, nous admettrons l'écriture de dérivation suivant plusieurs variables $\mathbf{v} = (v_1, \ldots, v_n)$.

$$\frac{\partial f}{\partial \mathbf{v}} = \left(\frac{\partial f}{\partial v_1}, \ldots, \frac{\partial f}{\partial v_n} \right)$$

Le gradient ∇ désigne les dérivées partielles suivant les coordonnées absolues :

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right) .$$

Dans le but de simplifier les notations, nous introduirons le cas particulier de dérivation par rapport à un vecteur de coordonnées concaténé $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_n) = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n) \in \mathbb{R}^{3n}$.
Soit la fonction scalaire $g : \mathbf{P} \mapsto g(\mathbf{P})$, son *vecteur* dérivé suivant les composantes x (resp. y, z) du vecteur $\mathbf{P}$ sera désigné par $\nabla_{\mathbf{x}} f$ (resp. $\nabla_{\mathbf{y}} f$, $\nabla_{\mathbf{z}} f$), avec

$$\nabla_{\mathbf{x}} f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right) ,$$

et de même pour y et z.

INTRODUCTION

AU cours de ces vingt dernières années, les modèles de personnages et objets de synthèse sont devenus partie intégrante des divertissements grands public. Sous l'impulsion du développement des films d'animation et des jeux vidéos, le besoin en modélisation, déformation et animation de modèle virtuel ne cesse de s'accroitre.

Cette thèse s'inscrit dans le contexte de la synthèse d'images, en particulier pour l'applicatif de l'industrie du loisir. Les outils professionnels d'édition 3D utilisés par les artistes numériques se doivent d'évoluer afin de pouvoir répondre à la demande croissante de modèles virtuels toujours plus nombreux et complexes comme par exemple illustrée en fig. 1. Dans



FIGURE 1 – *Modèles virtuels utilisés en production de film d'animation. À gauche plusieurs créatures virtuelles se déplaçant, bien qu'imaginaires, leurs mouvements et déformations restent plausibles (image issue du film* Le Seigneur des Anneaux : Le Retour du Roi*). À droite des personnages d'un film d'animation dont les vêtements présentent de nombreux détails animés (image issue du film d'animation* Shrek*).*

ce cadre, nous soulevons deux problématiques : Premièrement, ces modèles doivent à la fois revêtir une apparence réaliste à un instant donné, mais également posséder un comportement

plausible aux yeux du public lorsqu'ils sont animés. Deuxièmement, les artistes numériques qui génèrent ces modèles doivent pouvoir les manipuler interactivement, et ce, de manière intuitive.

PROBLÉMATIQUE

Notre perception humaine du réalisme est particulièrement sensible à l'apparence et au mouvement des objets qui nous sont familiers. Tout défaut par rapport à un comportement attendu nous donnera l'impression que le modèle entier est un *faux*.

La manière dont se déforment ces différents modèles est donc d'importance capitale pour donner l'impression d'un monde virtuel convaincant. Plus que la modélisation précise des déformations sous-jacentes, l'importance est ici donnée à l'apparence plausible des modèles visuels. Il est pour cela indispensable de respecter certaines contraintes pour une catégorie d'objet donnée. Par exemple, un personnage, dont l'épaule gonfle lorsqu'il marche, un vêtement se déformant sans plis, ou une feuille de papier s'étirant de manière élastique sont des exemples de déformations visuelles jugées non satisfaisantes pour le grand public.

Les outils professionnels d'éditions 3D utilisés par les artistes numériques ne fournissent cependant pas de méthodes de déformations satisfaisant de manière automatique ces contraintes géométriques lors de l'édition. De ce fait, les artistes numériques doivent ajuster et corriger manuellement de nombreux paramètres. Lors d'animation, ces paramètres sont de plus retravaillés à chaque pas de temps. Une grande partie du temps investi par les artistes n'est ainsi pas passée sur la création artistique elle-même, mais sur l'aspect technique de l'utilisation des outils 3D appliqués à des tâches fastidieuses d'ajustements. À l'heure actuelle, quelques secondes d'animation d'une production cinématographique peuvent représenter plusieurs semaines de travail intensif pour plusieurs artistes expérimentés.

MOTIVATION

Ce travail présente différents outils permettant de faciliter le travail de modélisation et d'animation de surfaces. Celles-ci, réagissent différemment suivant l'a-priori de leur constitution interne afin de préserver leurs caractéristiques propres indépendamment des contraintes extérieures. Nous proposons ainsi un modèle de surfaces que l'on désignera par le terme d'*actif*. Pour chaque application proposée, nous définissons une approche procédurale permettant de préserver des contraintes intrinsèques satisfaisant aux critères suivants :

- **Plausibilité** : Les contraintes géométriques maintenues par nos modèles actifs permettent de rendre les modèles virtuels visuellement plausibles par rapport à une modélisation ou une déformation ne les prenant pas en compte. Plus que la recherche de la précision numérique de la résolution des contraintes, nous mettons en avant des critères plus importants sur le plan visuel tels que la localité, ou la modélisation de discontinuités des plans tangents.
- **Efficacité** : Les approches décrites permettent la déformation et l'animation des modèles virtuels à vitesse interactive sur les machines grand public standard. Au-delà des problématiques d'implémentation, nous proposons des méthodes qui ne requièrent pas d'optimisation globale, mais plutôt basées sur des expressions locales analytiques intégrant directement les contraintes géométriques.
- **Contrôle** : Il est essentiel de garder un contrôle de haut niveau sur le résultat visuel d'une déformation géométrique. En effet, il est généralement souhaitable que les artistes numériques puissent manipuler la forme finale, ou la localisation de certaines déformations afin d'aboutir au résultat souhaité. L'utilisation des contraintes géométriques permet alors de garantir un résultat plausible sans pour autant limiter la créativité recherchée.
- **Utilisabilité** : Les différentes méthodes proposées sur des pipes-lines de production standard sont d'ores et déjà efficacement implémentées et utilisées dans les logiciels professionnels de modélisation 3D. L'utilisation de post-traitement en sus de méthodes telles que le skinning ou la simulation masse-ressorts permet l'utilisation nos approches pour améliorer le résultat obtenu sans devoir introduire de modification importante aux méthodes déjà mises en place.

CONTRIBUTIONS

Nos contributions se décomposent autour de trois axes principaux.

Notre premier axe de contributions s'articule autour d'une méthode de préservation de volume s'appliquant en temps réel sur des personnages *skinnés* [1]. Nous proposons une estimation rapide de volume local suivi d'une correction. Contrairement aux approches globales, notre contrôle du volume génère un gonflement local, dont la localisation est automatiquement calculée grâce à une combinaison de la contribution des poids de skinning et de la distance de la peau au squelette.

1. Nous utilisons ce terme tiré de l'anglais pour désigner un personnage habillé d'une peau géométrique qui suit les déformations de son squelette

Nous proposons également un calcul itératif exact du changement de volume induit localement par chaque os, suivi d'une correction du volume locale également exacte. Une nouvelle méthode de contrôle de la déformation est proposée grâce à l'utilisation de courbes de profils permettant d'imiter aisément différents effets tels qu'un muscle se gonflant ou des bourrelets. Nous introduisons également une orientation privilégiée de la déformation s'adaptant automatiquement à l'animation et permettant de réaliser des effets anisotropes comparables à la déformation visuelle d'un tube en caoutchouc. Afin de pouvoir s'appliquer directement sur des animations préexistantes, la méthode peut être utilisée de manière exacte sur des maillages triangulaires ainsi que des maillages quadrangulaires.
De plus, nous introduisons un nouveau calcul de poids de skinning basé sur une distance géodésique volumique permettant de générer des animations visuellement plausibles à partir de l'unique information d'un maillage et d'un squelette d'animation.

Un second axe de contributions se concentre sur l'animation rapide et réaliste des vêtements d'un personnage, mettant en avant le caractère quasi développable de ces surfaces.
Nous proposons pour cela une méthode automatique d'ajout de plis sur des animations de vêtements à basse résolution. Contrairement aux approches précédentes, notre méthode ne requière pas d'exemples d'apprentissages ni le tracé manuel de la localisation et de l'orientation de ces plis.
Nous introduisons également une nouvelle méthode de contrainte temporelle rapide permettant de faire apparaître, disparaître, ou simplement glisser continument les plis sur la surface du tissu en mouvement.
Enfin, nous introduisons des déformeurs implicites permettant de générer la géométrie finale des plis se greffant sur la surface d'origine. Ces plis peuvent de plus, tout comme dans le cas réel, se rapprocher les uns des autres et finir par se mélanger de manière lisse pour donner l'apparence d'un unique pli sans générer d'auto-intersection. Nous traitons ainsi de manière transparente pour l'utilisateur des topologies de plis quelconques, pouvant s'adapter dynamiquement lors de l'animation, tout en assurant l'absence d'auto-intersection de manière robuste.

Un dernier axe de contribution concerne la modélisation efficace de surfaces ayant l'apparence de papier froissé. À partir de l'information du bord sur lequel doit s'appuyer la surface finale, nous générons de manière rapide une surface plissée s'appuyant sur ces positions, tout en préservant au mieux l'isométrie par rapport à son patron originel. La surface générée est obtenue rapidement sans nécessiter d'optimisation globale. De plus, elle gère de manière native les discontinuités des plans tangents en engendrant des lignes de plis, qui sont ensuite suivies explicitement. Contrairement à l'utilisation des simulations physique actuelles, la surface finale peut ainsi posséder des plis francs caractéristiques de l'apparence du papier froissé.

Organisation du manuscrit

Ce manuscrit s'organise en quatre parties.

Nous réaliserons dans un premier temps un état de l'art des différentes méthodes de modélisation et d'animation de modèles virtuels. Le chapitre 1 présentera ainsi les différentes représentations de surface possibles pour l'animation et la modélisation (section 1.1), ainsi que les deux grandes approches de déformation : approches physiques (section 1.2) et géométriques (section 1.3). Enfin, nous passerons en revue les approches existantes de déformation préser-

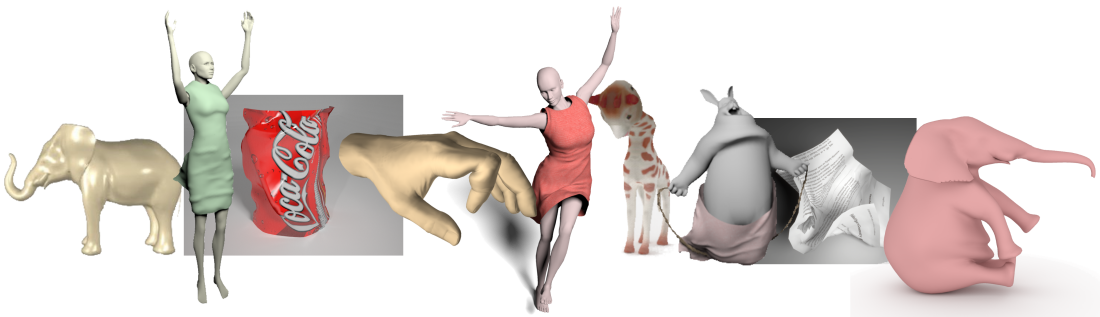
vant certaines contraintes intrinsèques telles que le volume ou les longueurs en expliquant leurs limitations actuelles (section 1.4).

Le chapitre 2 présentera nos contributions concernant l'animation de créatures virtuelles articulées. Nous considérerons en particulier le cas de la déformation par skinning avec l'introduction d'une méthode de calcul de poids de skinning (section 2.1), le développement d'un estimateur local du changement de volume lors de l'animation (section 2.2), et d'une nouvelle méthode de déformation secondaire permettant de contrôler à la fois le volume (section 2.3) et la forme souhaitée (section 2.4). Après avoir comparé les résultats des différentes approches présentées (section 2.5, 2.6), nous appliquons et étendons celles-ci dans le cadre d'un outil de réalité virtuelle (section 2.7), ainsi qu'à d'autres contraintes géométriques telles que la préservation d'aire (section 2.8).

Le chapitre 3 développera nos contributions concernant la génération rapide de plis pour l'animation de vêtements quasi incompressibles. Nous expliciterons pour cela une localisation automatique de courbes de plis (section 3.1), ainsi que des contraintes temporelles permettant une déformation continue de ces courbes (section 3.2). Nous détaillerons ensuite la déformation géométrique associée aux plis (section 3.3).

Le chapitre 4 concerne notre contribution à la modélisation de surfaces ayant l'aspect de papier froissé. Nous présenterons une méthode de génération de données d'entrées compatibles avec la modélisation de telles surfaces (section 4.1). Puis nous développerons un algorithme de subdivision récursif associé à une relaxation globale améliorant la développabilitée (section 4.2), avant de montrer que la méthode s'adapte de manière robuste à différents modèles de surface (section 4.3).

Finalement, nous conclurons sur les différentes approches décrites et proposerons des pistes d'améliorations futures.



CHAPITRE

1

ÉTAT DE L'ART SUR LES MÉTHODES DE DÉFORMATION DE SURFACES

LA modélisation par surface est, comme nous allons le voir, une représentation essentielle en modélisation et en animation. Le choix de la représentation d'une surface n'est pas sans conséquence sur la manipulation de l'objet sous-jacent. Il est donc important de considérer des modèles de représentation et de déformations adaptés afin de pouvoir les manipuler aisément et efficacement.

Après avoir expliqué le choix de l'utilisation de structure d'une surface, nous détaillerons différentes possibilités de représentation de celle-ci. Nous réalisons ensuite un état de l'art sur les différentes approches de déformations de surfaces. Pour cela, nous expliquerons les avantages et inconvénients de l'utilisation de méthodes de déformations dites *physiques*. D'un autre côté, nous détaillerons certaines méthodes de déformations purement géométriques, et enfin, nous considèrerons les applications déjà existantes de préservation de propriétés intrinsèques des surfaces.

1.1 DÉFINITION ET NOTATIONS DE LA REPRÉSENTATION PAR SURFACES

Motivation de la représentation par surface :

Les objets qui nous entourent sont intrinsèquement constitués de matière volumique se répartissant dans les trois dimensions spatiales. Que ce soit les muscles situés à l'intérieur d'un corps humain, ou des fibres d'une feuille de papier, la structure volumique constitutive des objets et êtres vivants est à l'origine des caractéristiques mécaniques de leurs déformations.

Ceci étant, si ces structures volumiques restent essentielles à une modélisation de la physique du matériau, elles n'entrent pas en compte dans l'apparence visuelle d'un matériau non transparent comme illustré en fig. 1.1. La modélisation d'un volume complet est de plus ex-

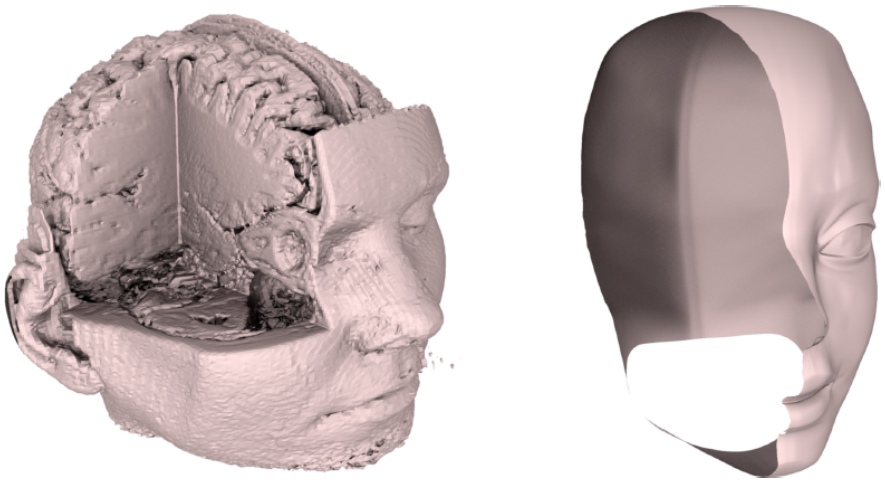


FIGURE 1.1 – *Gauche : Représentation volumique d'une tête humaine. Droite : Représentation surfacique. La représentation volumique contient une plus grande quantité d'information que l'approche surfacique. Cette dernière est cependant suffisante pour représenter l'aspect extérieur de la tête.*

cessivement lourde à modéliser et à gérer d'un point de vue complexité, temps de calculs et espace mémoire. Pour cela, la majeure partie des modélisations des formes et objets de synthèse se réalise uniquement par rapport à la partie visible de ceux-ci, c'est-à-dire leurs bords surfaciques.

Cette thèse s'articulant autour de la modélisation et l'animation rapide d'objets de synthèse, nous allons considérer de manière générale qu'une forme est définie par sa surface frontière. Certains objets que l'on recherchera à modéliser seront considérés d'épaisseurs nulles (ex. tissus, papiers, ...). D'autres par contre seront considérés comme possédant un intérieur délimité par la surface bien que celui-ci ne sera pas modélisé de manière explicite (ex. personnages et animaux).

Si cette représentation surfacique suffit à elle seule pour la visualisation de l'objet, nous n'oublions pas que la déformation et l'animation de celui-ci sont guidées par des propriétés venant de leurs structures physiques volumiques. Notre approche consiste donc à considérer le résultat de ces propriétés physiques comme s'appliquant de manière géométrique sur la surface visible. Nous proposons ainsi d'extraire certaines contraintes géométriques à assurer sur la surface finale, afin de modéliser le comportement plausible d'un objet réel. La surface finale peut alors être manipulée par des outils de déformation classiques, mais sera contrainte à maintenir certaines des propriétés intrinsèques à l'objet sous-jacent. Nous désignerons ce type de surface par *active*.

Modélisation d'une surface :

Une surface est une abstraction géométrique qui dans notre cas est utilisée pour représenter la forme d'un objet virtuel. La surface définit donc la frontière physique d'un objet dans l'es-

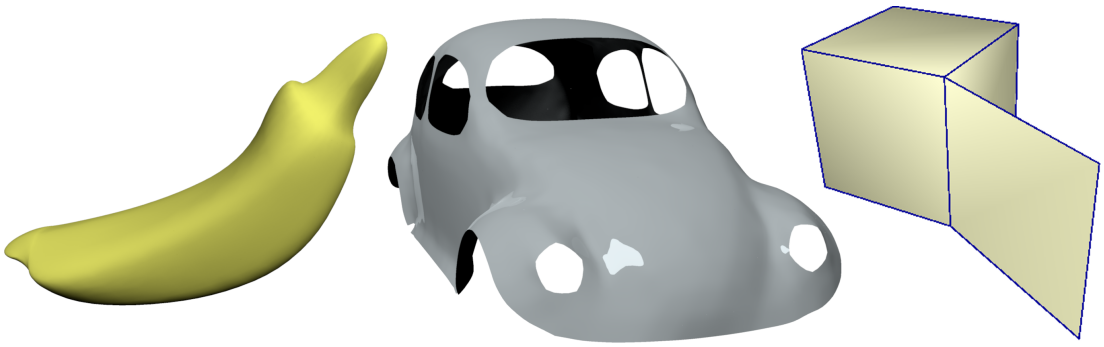


FIGURE 1.2 – *Exemples de différents types de surfaces possibles. Gauche : 2-variété sans bord, tout voisinage d'un point est un disque complet. Milieu : 2-variété à bords, tout voisinage d'un point est un disque ou un demi-disque sur le bord. Droite : surface non variété, le voisinage à la jonction avec le plan étant un disque plus un demi-disque.*

pace. C'est elle qui contient les propriétées fondamentale de la forme. Une fois la géométrie définie, il reste important d'attacher d'autres paramètres sur cette forme (couleurs, textures, ...) afin d'obtenir un résultat visuel convainquant. Dans un premier temps, nous allons nous intéresser uniquement à la forme géométrique et décrire comment une surface peut être représentée et manipulée en informatique.

On rappelle que d'une manière générale dans ce document, nous nous placerons toujours dans l'espace Cartésien $\mathbb{R}^3$ muni de son repère canonique, et une position dans l'espace par rapport à l'origine O sera désignée par le vecteur $\mathbf{p} = (x, y, z)$.

Formellement, nous utiliserons le terme de surface pour désigner un sous ensemble de $\mathbb{R}^3$ 2-variété orientable (voir fig. 1.2). C'est-à-dire que le voisinage topologique de tout point de la surface est homéomorphe [1] à un disque (ou à un demi-disque le long d'un bord). Nous considérons également que nos surfaces représentent des objets réels plongés et orientables.

1.1.1 Modélisation explicite

La première solution pour modéliser une surface consiste à définir de manière explicite l'ensemble des positions de l'espace par laquelle la surface passe. Nous verrons tout d'abord la définition explicite d'une surface de manière globale. Une définition globale d'une surface complexe étant, sauf cas très particulier, non exprimable mathématiquement, nous montrons par la suite comment approcher ces formes complexes par des modèles discrets.

Surface paramétrique

Dans un premier temps, nous définisson une surface S dans sa globalité en exprimant paramétriquement l'ensemble des positions par laquelle elle passe. Soit $\mathcal{D} \subset \mathbb{R}^3$ un domaine du plan associé à un système de coordonnées (u, v) paramétrant ce domaine. Soit l'application f

1. transformable continument sans déchirure ni soudure, c'est à dire issue d'une transformation bijective continue dont la réciproque est continue également

par

$$f : \left\{ \begin{array}{ccc} \mathcal{D} \subset \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ (u,v) & \mapsto & f(u,v) \end{array} \right.$$

La surface S est l'image de l'application f dans $\mathbb{R}^3$ (voir fig. 1.3), c'est à dire

$$S = \left\{ \mathbf{p} \in \mathbb{R}^3 \mid \exists (u,v) \in \mathcal{D} \ , \ \ \mathbf{p} = f(u,v) \right\}$$

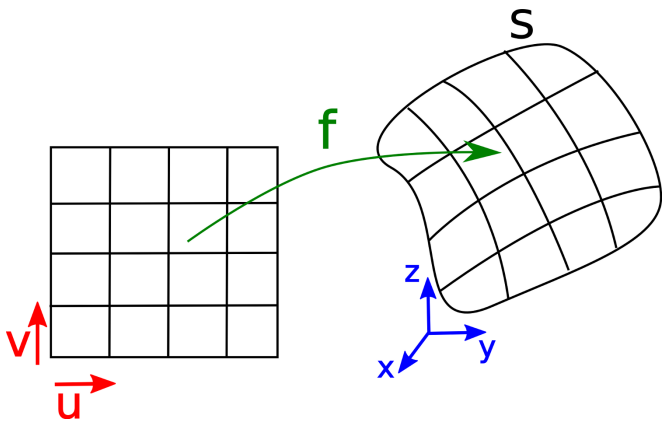


FIGURE 1.3 – *Exemple de mapping f de l'espace 2D vers 3D. L'image de f dans $\mathbb{R}^3$ représente la surface S.*

Une surface peut donc être vue comme le résultat de la déformation d'un domaine $\mathcal{D} \subset \mathbb{R}^2$ vers un sous-ensemble de $\mathbb{R}^3$, et l'application f peut ainsi être considérée comme une fonction de déformation de l'espace 2D vers 3D. On désignera ainsi dans la suite du document cette application par le terme spécifique de *mapping*.

Cette fonction f contient les informations d'une paramétrisation globale définie par ces paramètres (u,v). Bien qu'il soit classique de confondre abusivement le mapping f avec la surface S, les deux ne sont pas équivalents. Plus précisement, la surface S constitue la trace du mapping f dans $\mathbb{R}^3$. La connaissance de f implique celle de S. La réciproque étant fausse, et de plus f n'est pas unique pour une surface donnée.

Nous avons vu que la connaissance de f contient en fait plus d'informations que la simple structure géométrique de la surface en encodant une paramétrisation globale. Malheureusement, cette paramétrisation est, sauf cas exceptionnels, impossible à exprimer exactement par les fonctions mathématiques usuelles. Il est par contre d'usage en informatique graphique de définir des modèles discrets de surface. C'est-à-dire que la position, ainsi que des paramètres de continuités de la surface ne sont connus exactement qu'en un nombre fini de points, le reste étant comblé par interpolation ou approximation. On défini alors la surface non plus globalement, mais par morceaux. Nous détaillons par la suite diverses approches permettant de définir une surface complète par morceaux.

Modèle discret par morceaux

A Modèle paramétrique

D'une manière générale, la modélisation par morceaux des surfaces consiste à définir un ensemble discret de mappings $(f_i)_{i\in[[1,N]]}$ (polynomiaux pour la plupart) défini sur des domaines $\mathcal{D}_i$ tel que l'union respective de leurs images S_i dans l'espace représente une surface $S = \bigcup S_i$.
Cette composition par morceaux permet de traiter les caractéristiques locales séparément, rendant la modélisation plus aisée. Dans la vaste majorité des cas, définir exactement une surface donnée à l'aide d'un nombre fini de morceaux est très difficile voir impossible. Il est néanmoins possible d'approcher avec une erreur contrôlée le degré d'approximation de celle-ci. Un choix adéquant de modélisation permet dans certains cas de faire converger ce modèle discret vers la vraie surface lorsque le nombre de morceaux tend vers l'infini.
Afin de rendre la tâche plus aisée, il est classique de considérer des mappings f_i définis sur un même domaine $\mathcal{D}$ triangulaire ou quadrangulaire indépendant du morceau. Cela revenant à accoler N patchs initialement identiques dans le plan mais déformés séparément comme illustré en fig. 1.4. Il reste ensuite à définir la fonction de déformation f_i pour un morceau donné. En fonction du degré d'approximation et des contraintes à respecter, différents choix sont possibles.

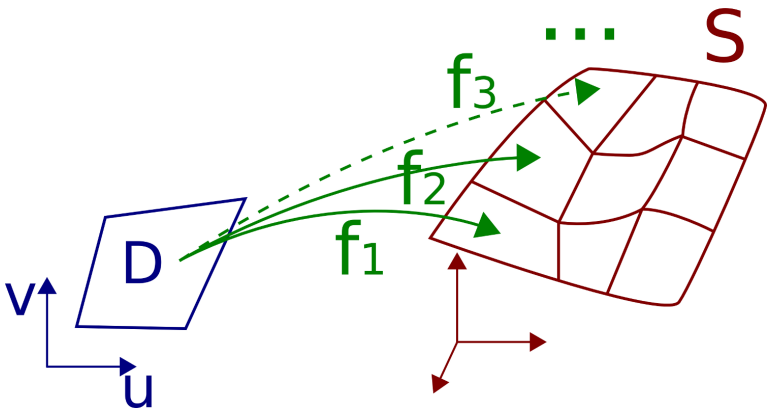


FIGURE 1.4 – *Exemple de surface S définie par morceaux à partir d'un même domaine $D \subset \mathbb{R}^2$.*

i Approximation d'ordre 1

Considérons le cas le plus simple où $\mathcal{D}$ est donné par le triangle canonique du plan

$$\mathcal{D} = \{(u, v) \in \mathbb{R}^+ \times \mathbb{R}^+ \mid u + v < 1\}.$$

On définit alors f_i par un mapping linéaire. L'image de $\mathcal{D}$ est alors une surface triangulaire plane de $\mathbb{R}^3$. On peut alors écrire

$$f_i : \begin{cases} \mathcal{D} \subset \mathbb{R}^2 & \to \mathbb{R}^3 \\ (u, v) & \mapsto f_i(u, v) = \mathrm{L}_i \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \end{cases},$$

avec L_i une matrice 3×3 constante sur le morceau i considéré.
Définissons $(\mathbf{x}_0^i, \mathbf{x}_1^i, \mathbf{x}_2^i)$ les trois positions spatiales du triangle image de $\mathbb{R}^3$ (voir fig. 1.5). En remarquant que ces trois positions sont les images des extrémitées respectives du triangle du plan définies par $(u, v) = (0, 0)$, $(1, 0)$ et $(0, 1)$, il est possible d'exprimer L_i en fonction des coordonnées 3D par

$$L_i = \left(\mathbf{x}_1^i - \mathbf{x}_0^i,\ \mathbf{x}_2^i - \mathbf{x}_0^i,\ \mathbf{x}_0^i\right)^T$$

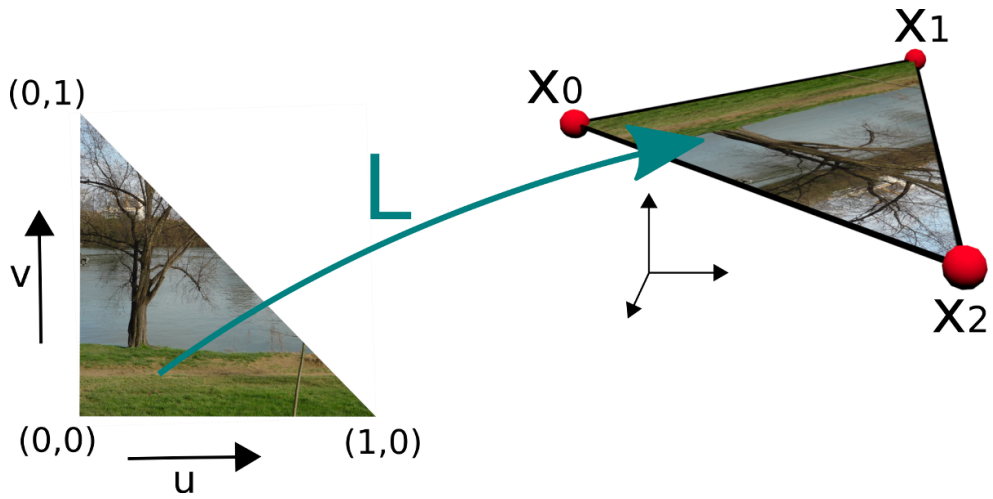


FIGURE 1.5 – *Déformation linéaire appliquée sur un triangle.*

D'une manière générale, la surface S est donc constituée d'un ensemble de triangles dans l'espace. On parle alors de maillage triangulaire. Il s'agit de la forme la plus simple pour représenter une surface. Voici quelques avantages et inconvénients :

Avantages
- Peut représenter tout type de formes, la structure triangulaire laisse toute liberté en terme de position et d'organisations des positions discrètes. Plus précisément, il est possible de modéliser une 2-variété de topologie quelconque.
- Il s'agit de la manière la plus simple d'un point de vue théorique et la plus rapide de représenter une forme.
- Toute représentation de classe supérieure peut trivialement s'approximer par un ensemble de triangles.

Inconvenients
- Ordre d'approximation le plus bas. Pour une fonction f lisse, c'est donc une approximation de faible qualité. Cette représentation nécessite ainsi l'utilisation d'un échantillonnage discret très fin pour approximer la surface sous-jacente comme illustrée en fig. 1.6.
- f est globalement C^0, mais jamais C^1 sauf dans le cas d'un plan. La géométrie de la surface peut ainsi être continue, mais elle présente des arêtes vives. Une surface lisse non plane ne pourra jamais être exactement représentée.

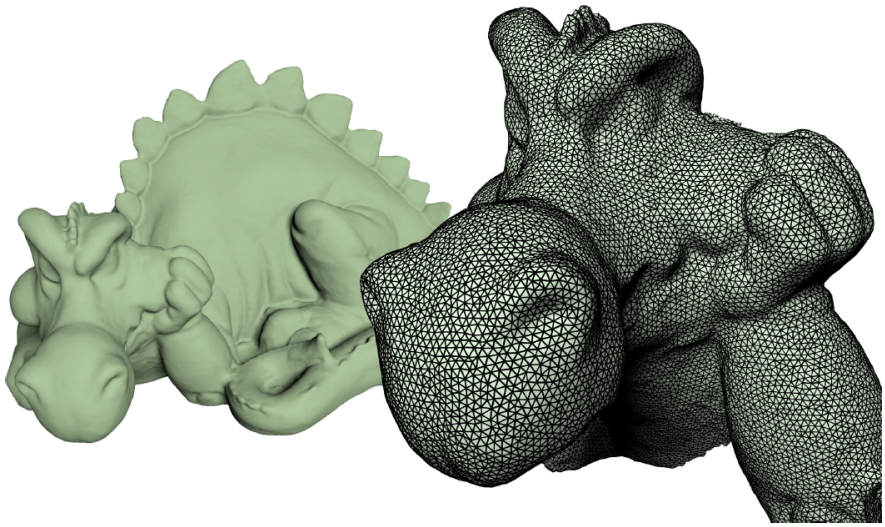


FIGURE 1.6 – *Exemple d'approximation d'une surface complexe par un maillage triangulaire. La représentation nécessite un nombre très important de triangles pour modéliser l'ensemble des détails.*

ii **Approximations polynomiale d'ordre supérieur**

– Surface B-Spline produit tensoriel

Afin de pallier à la faible qualité de l'approximation linéaire par morceaux, de nombreux travaux se proposent de traiter des cas où f est une fonction polynomiale de degré > 1. Les polynômes se généralisent en multi-dimension (ici 2) par le produit tensoriel de polynômes de degrés 1. Dans le cas des surfaces, nous nous retrouvons avec des polynômes de degrés arbitraires obtenus par produit tensoriel de polynômes unidimensionnels de plus hauts degrés.
L'approximation résultante est meilleure que dans le cas linéaire, et permet de modéliser des surfaces lisses avec un nombre plus faible de morceaux de surface. On citera en particulier le cas de l'utilisation de polynômes bi-linéraires, et bi-cubiques avec leurs déclinaisons en fonctions de bases splines permettant d'obtenir des continuités de tangentes et de courbures localement contrôlables, voir [Far01] pour plus de détails.
Ces modélisations sont cependant plus lourdes à gérer et ne sont pas adaptées à la représentation de topologies quelconques. En effet, leur paramétrisation sous-jacente rectangulaire dites en *jeu d'échec* nécessite l'introduction de deux directions principales de manière globale. De plus, elles se prêtent mal à des modifications locales telles que des ajouts de détails non prévus initialement par la grille de contrôle.

– Surface splines de topologie quelconque

Initiées par Forsey *et al.* [FB88], puis étendues par Gonzalez-Ochoa et Peters [GOP99], les surfaces splines de topologie quelconque peuvent néanmoins palier à ce défaut lorsque la structure de contrôle satisfait une régularitée suffisante à travers l'ensemble des niveaux de la hiérarchie de subdivision. Plus tard, Yvart *et al.* [YHB05] proposent une approche hierachique s'adaptant sur des splines à support triangulaire. Ils permettent ainsi de s'adapter au cas répandu de surfaces

triangulaires de topologie quelconque tout en permettant d'accéder à des degrés de continuités G^1.
Bien que potentiellement adaptables à des topologies arbitraires, ces représentation restent peu implémentées dans les logiciels de modélisation graphiques utilisées par les artistes numériques de par leur manipulation plus complexe. Ces surfaces restent principalement utilisées dans le domaine de la CAO.

B Cas particulier du modèle polygonal

Nous avons pu voir que les surfaces définies par portions planes triangulaires étaient les plus aisées à manipuler.
Un grand nombre d'outils de modélisation professionnels proposent de plus la gestion native des surfaces triangulaires, voir plus généralement de maillages polygonaux.
La partie suivante développe quelques caractéristiques spécifiques à cette représentation dont nous nous servirons régulièrement. D'avantage de détails sont disponibles dans [BPK+07, BKL+09].

i Structure

Un maillage polygonal $\mathcal{M}$ est un ensemble

$$\mathcal{M} = (\mathcal{F}, \mathcal{E}, \mathcal{V}) \,,$$

où $\mathcal{V} = (v_1, \ldots, v_{N_s})$ représente les N_s sommets du maillage,

$$\mathcal{F} = (f_1, \ldots, f_{N_p}) \in (\mathcal{V} \times \mathcal{V} \times \mathcal{V})^{N_p} \,,$$

représente les N_p faces et

$$\mathcal{E} = (e_1, \ldots, e_{N_e}) \in (\mathcal{V} \times \mathcal{V})^{N_e} \,,$$

représente les N_e arêtes (voir fig. 1.7).

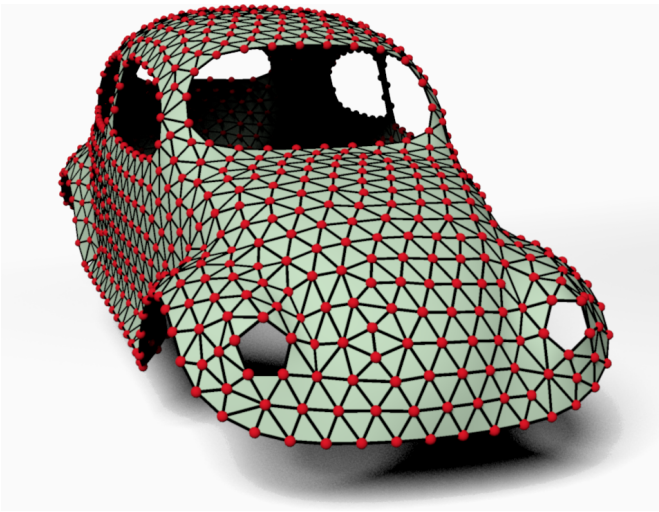


FIGURE 1.7 – *Structure d'un maillage représentant une carrosserie. Les sommets sont représentés par les sphères rouges, les arêtes par les segments noirs, et les faces (ici triangles) en vert.*

ii **Connectivité et plongement géométrique**
La structure précédente représente la connectivité du maillage, et est donc indépendante du plongement de la forme dans l'espace.
En associant l'application qui à chaque sommet $v_i \in \mathcal{V}$ associe les coordonnées 3D dans $\mathbb{R}^3$, on obtient alors l'information de géométrie du maillage portée par le plongement de $\mathcal{V}$.

iii **Structure de données**
Il est commun en informatique graphique d'utiliser une structure de données spécifique pour la manipulation de maillages. On parle généralement de structure B-Rep (Boundary Representation). Notons qu'il existe plusieurs structures de données possibles (indexée [THLR98], basée arêtes [CGA, Ope], cartes [Lie94], ...). Nous détaillons la notation indexée sur laquelle nous baserons nos descriptions ultérieure.

L'ensemble des sommets est stocké par le vecteur des coordonnées spatiales. On a alors $\mathcal{V} = (\mathbf{p}_1, \ldots, \mathbf{p}_{N_s}) \in (\mathbb{R}^3)^{N_s}$.
L'ensemble des faces $\mathcal{F}$ contient N_p uplets d'indices désignant le numéro de sommet de $\mathcal{V}$ rencontré lors du parcours le long de l'arête d'un polygone donné.
$\mathcal{E}$ contient N_e couple d'indices correspondants aux sommets de $\mathcal{V}$ de chaque arête.

Notons que dans le cas de déformations, la modification de $\mathcal{V}$ uniquement ne permet pas de modifier la topologie de la surface (la surface peut par contre posséder des auto-intersections). La modification du voisinage d'un point passe forcément par une modification de la connectivité $(\mathcal{F}, \mathcal{E})$.

iv **Cas particuliers de maillages, cas des modélisations par les artistes**
Un cas particulier très fréquent concerne les maillages triangulaires. $\mathcal{F}$ est alors uniquement formé de N_p 3-uplets. Un autre cas rencontré classiquement lors de la manipulation de formes générées par des artistes concerne les maillages quadrangulaires. La modélisation par quadrangles permet de définir des lignes caractéristiques principales. Ces directions principales permettent notamment d'aider à séparer différentes parties mobiles lors d'animations. Dans ce cas, le polygone associé est le quadrangle [2], ou *quad*.
Modéliser une surface entièrement à partir de quadrangles reste cependant complexe, voir impossible selon les cas. Dans le cas général, les maillages créés par les artistes sont ainsi formés d'un mélange de quadrangles et de triangles. Notons qu'un ensemble de quadrangles peut toujours se trianguler trivialement. Des exemples sont montrés en fig. 1.8.

1.1.2 Modélisation implicite

Jusqu'à présent, nous avons vu comment définir une surface de manière explicite. Cette définition explicite est avantageuse pour le rendu de la surface. Cette représentation ne donne par contre pas d'information par rapport à la position relative d'un point situé hors de la surface, et demande la gestion explicite de la connectivité. Les changements topologiques parfois nécessaires lors d'une animation demandent ainsi d'être traités explicitement ce qui se révèle complexe d'un point de vue algorithmique et couteux en temps de calcul.

2. Notons que ces quadrangles peuvent être plan ou non. Dans ce dernier cas, la visualisation se réalise généralement par triangulation, ou par interpolation bilinéaire.

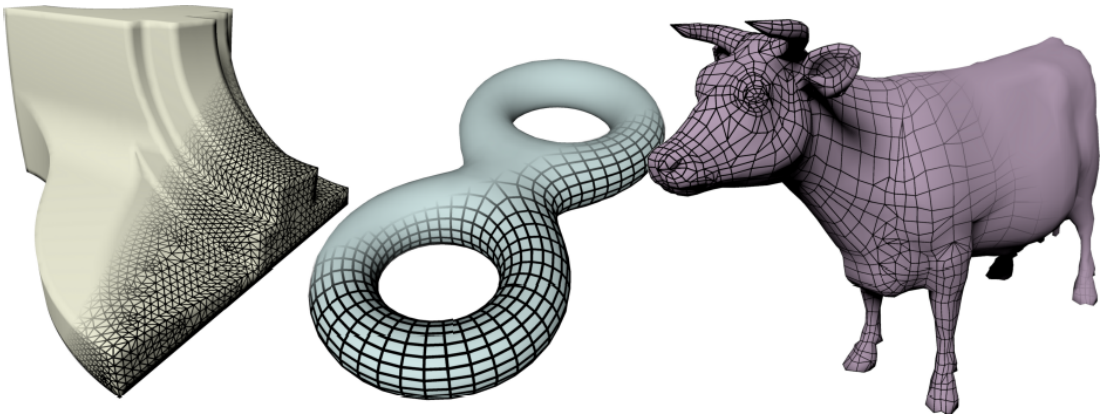


FIGURE 1.8 – *Exemple de maillages. Gauche : Maillage triangulaire. Milieu : Maillage formé de quadrangles. Droite : Maillage mixte.*

Une autre représentation permet quant à elle de traiter de manière transparente les modifications de topologie. Il s'agit de la définition implicite de la surface.

Modèle continu

Soit S une surface plongée dans $\mathbb{R}^3$. Définir implicitement S revient à caractériser l'ensemble des positions décrites par la surface comme étant solution d'une équation à 3 variables. On a donc

$$S = \{\mathbf{x} \in \mathbb{R}^3 | F(\mathbf{x}) = \text{iso}\}\ , \tag{1.1}$$

où iso est une valeur scalaire constante.

$F : \mathbb{R}^3 \rightarrow \mathbb{R}$ étant dénomé potentiel ou fonction de champ, et iso étant une valeur scalaire communément appelée l'isovaleur. Un exemple est donné en fig. 1.9.

Notons que pour S donné, le choix de F n'est pas unique. De plus, toute couple (F, iso) ne définit pas une surface telle que décrite précédemment par le mapping d'un domaine du plan dans l'espace. Certains couples pouvant définir des cas dits dégénérés : l'ensemble vide, un ensemble de positions isolées de l'espace, ou un domaine volumique. Notons que l'on peut également obtenir un mélange de ces différents cas.

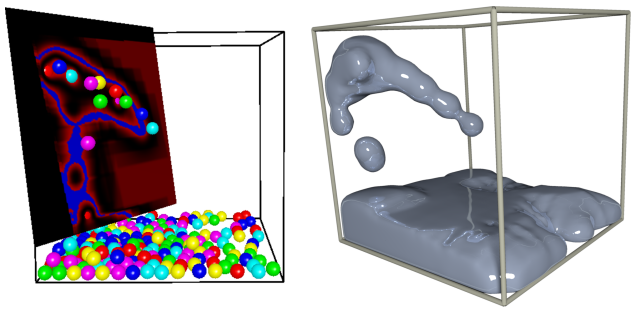


FIGURE 1.9 – *Exemple de surface implicite. Une fonction volumique vue en coupe est définie à partir de particules discrêtes (gauche). L'extraction de l'isosurface permet de donner l'impression d'une surface d'un fluide (droite).*

Ces nombreuses possibilitées traduisent un plus grand degré de liberté issu de la définition, qui ne nécessite pas l'encodage de la connectivité entre les positions des points de la surface.

Voici certaines différences notables entre une modélisation implicite et explicite :

Avantages
- Pas de gestion directe de la topologie de la surface. Celle-ci est entièrement définie par le couple (F, iso).
- Une modification de l'unique valeur scalaire iso à F fixé permet de définir un ensemble de surfaces qui ne s'intersectent pas, et ayant potentiellement des topologies différentes.

Inconvénients
- Il n'est pas possible de connaitre l'ensemble des positions situés sur la surface directement. On ne peut que savoir qu'à postériori si une position spatiale appartient ou non à la surface.
- Étant donné un point de la surface, on ne connait pas son voisinage explicitement sur la surface.

Conséquence : Les surfaces implicites sont les modèles de prédilection pour traiter des surfaces changeant de topologies. Elles ne sont par contre pas directement visualisable, de par leur définition implicite.

De manière identique à la modélisation explicite, la connaissance exacte de la fonction F permettant de modéliser entièrement une surface complexe est généralement hors de portée. Il est donc nécessaire de passer par une étape d'approximation discrète. Plusieurs modèles discrets sont décrits-ci après.

Modèle discret voxelisé

Le modèle trivial de discrétisation consiste à échantillonner la fonction potentiel F dans l'espace. Généralement, cet échantillonnage se réalise dans une grille régulière. La boite unitaire située entre 4 échantillons spatiaux est appelée voxel ou élément de volume.

Une fois échantillonnée, une fonction F est définie dans tout l'espace par des méthodes d'interpolation ou d'approximation. Classiquement, on utilise l'interpolation tri-linéaire ou une approximation suivant des splines volumiques.

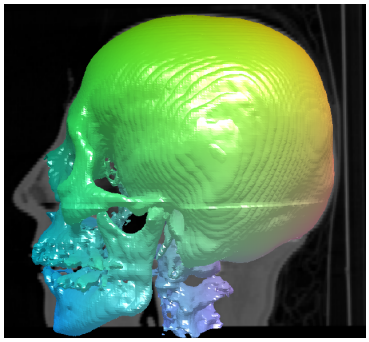


FIGURE 1.10 – *Exemple de données médicales stockées dans une grille 3D. Potentiel=image à niveaux de gris. Squelette=isosurface de ce potentiel.*

Étant donné un schéma d'interpolation, il est possible de retrouver la surface S en passant en revue l'ensemble des voxels de manière indépendante. Les méthodes classiques de reconstruction de surfaces maillées à partir de l'information discrétisée du potentiel sont le *Marching-Cube* [LC87] et le *Marching-Tetraèdre* [GH95].

Ce modèle voxelisé est le schéma d'approximation le plus généraliste et n'impose aucune contrainte sur la fonction. Par contre, il reste excessivement lourd à gérer en terme d'espace mémoire. En effet, il est nécessaire d'échantillonner l'espace finement si l'on souhaite modé-

liser des surfaces détaillées. Cela rajoute une dimension supplémentaire en terme de stockage qui reste difficilement gérable pour de grands volumes de données. On pourra noter qu'il s'agit du modèle de prédilection pour des applications médicales afin de traiter des données issues de scanner comme le montre la fig. 1.10.

D'autres approches plus spécifiques permettent d'éviter le problème d'une discrétisation totale de l'espace, et ainsi de permettre un traitement adapté et moins gourmand en terme de quantité de données à stocker.

Modèle discret à squelette

L'idée des modèles implicites à squelettes consiste à décomposer le champ de potentiel final F suivant une base de champs plus simple f_i. Ces champs de base définissent des *primitives* caractéristiques que l'on va venir mélanger implicitement. On a alors

$$F(\mathbf{p}) = \bigoplus_i f_i(\mathbf{p}) \, . \tag{1.2}$$

$\bigoplus$ est appelé l'opérateur de mélange. L'opérateur le plus courant étant la sommation direct des champs, mais il est possible de définir des opérateurs dédiés plus complexes souvent restreint aux combinaisons binaires.

Les fonctions f_i sont définies à partir d'une source manipulée explicitement. Cette source, appelée squelette est généralement constituée de points, de segments ou de triangles. Ainsi, la fonction associée possède des caractéristiques contrôlables liées au positionnement et à l'orientation du squelette le générant.

A. Exemple du cas des *Blobs* :

Le cas le plus simple et le plus répandu étant de considérer f_i comme étant une fonction décroissante de la distance au squelette. Supposons par exemple un squelette formé d'une seule position $\mathbf{p}_0$. La fonction de potentielle associée à ce squelette en une position $\mathbf{p}$ peut être donnée par

$$f_i(\mathbf{p}) = \exp\left(-\|\mathbf{p} - \mathbf{p}_0\|^2\right) \, . \tag{1.3}$$

La primitive implicite associée à ce type de définition est une sphère. Si au moins deux squelettes ponctuels sont spatialement proches, les champs vont se mélanger de manière lisse. Les surfaces associées à une isovaleur donnée peuvent alors varier continument entre deux formes sphériques séparées et une seule surface connexe lorsque les squelettes se rapprochent comme illustré en fig. 1.11. Ce type de représentation est connue sous le terme de *Blobs* [Bli82, Mur91].

B. Fonction de forme :

Il existe différentes possibilités pour définir l'évolution du potentiel à partir d'un squelette donnée. La fonction unidimensionnelle κ prenant comme variable la distance entre le point d'évaluation $\mathbf{p}$ et le squelette est appelée *fonction de décroissance* ou *fonction de forme* :

$$\kappa : \left\{ \begin{array}{ccc} \mathbb{R} & \rightarrow & \mathbb{R} \\ \|\mathbf{p} - \mathbf{p}_0\| & \mapsto & f_i(\mathbf{p}) \end{array} \right. \, . \tag{1.4}$$

Elle est caractéristique de la forme de la primitive implicite ainsi que de son rayon d'action. On notera qu'en modifiant cette fonction 1D, on peut ainsi définir un contrôle sur la forme finale de l'objet 3D.

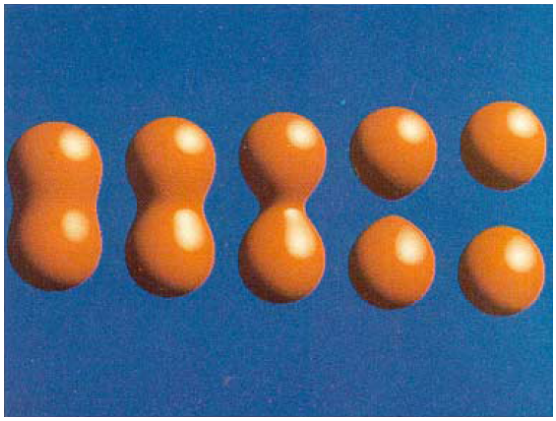


FIGURE 1.11 – *Illustration de deux surfaces issues d'une représentation implicite se mélangeant lorsque les centres respectifs des sphères sont spatialement proches. Illustration tirée de [Bli82].*

L'utilisation d'une fonction de décroissance exponentielle permet de définir des fonctions de classe C^∞, mais n'est pas à support compact. Ainsi, toute modification sur un squelette revêt un caractère global. On notera également la possibilitée d'utiliser des fonction de distance inverse permettant une évaluation rapide du champs de potentiel [WvO96].

L'utilisation de polynômes définis par morceaux sur un support compact permet d'adresser ce problème de localité comme dans le cas des *Metaball* [NHK+89], *Soft objects* [WMW86], ou encore *W-shaped polynomial* [PR], mais les fonctions ne sont plus aussi lisses.

C. Surfaces de convolution :

L'utilisation de fonctions de formes basées sur la distance est naturelle dans le cas de squelettes ponctuels. Il est cependant possible d'utiliser des squelettes 1D, ou même surfaciques pour modéliser des primitives implicites plus complexes. Sans perte de généralité, considérons le cas unidimensionnel d'un squelette formé par une courbe unidimensionnelle. Les primitives géométriques associées à ce type de squelette sont donc des cylindres généralisés pouvant se mélanger les uns par rapport aux autres.

Une courbe peut être vue comme la somme géométrique d'une infinité de positions spatiale. Par analogie, le champ résultat peut donc être considéré comme la somme infini des contributions des positions infinitésimales. La fonction f_i *naturellement* associé à un squelette unidimensionnel est donc obtenue par l'intégration des contributions infinitésimales d'une fonction de décroissance. Cette fonction κ est ainsi utilisée comme un noyau d'intégration, et est désignée par le terme de *fonction noyau*. On a alors pour une courbe $\mathcal{C}$

$$f_i(\mathbf{p}) = \int_{\mathbf{q}\in\mathcal{C}} \kappa(\|\mathbf{p}-\mathbf{q}\|)\,\mathrm{d}\mathbf{q}\,. \tag{1.5}$$

Introduites par Bloomenthal et Shoemake [BS91], les surfaces de convolutions permettent de générer des mélanges lisses de primitives implicites complexes à l'aide d'un mélange par sommation.

En effet, en utilisant la linéarité de l'intégrale vis-à-vis d'un mélange entre N primitives iden-

tiques, on peut écrire

$$F(\mathbf{p}) = \sum_i f_i(\mathbf{p}) = \int_{\mathbf{q}\in\mathcal{C}} \left(\sum_{i=1}^{N} \kappa\left(\|\mathbf{p}-\mathbf{q}_i\|\right) \right) \mathrm{d}\mathbf{q}_i \,, \tag{1.6}$$

où le terme situé après l'intégrale représente le potentiel issu de l'ensemble du squelette.

On en conclut que le mélange de plusieurs primitives implicites considérées indépendamment possède les mêmes propriétés que la fonction implicite issue de l'ensemble du squelette (voir un exemple en fig. 1.12).

Il en résulte deux avantages lorsque l'on considère κ suffisamment lisse :

1. La surface résultante d'une surface de convolution est également lisse (au même titre que la fonction κ), même si le squelette possède des discontinuités de tangentes.
2. La surface résultante ne possède pas de marques visuelles de transition [Blo97] lors du passage d'un squelette à l'autre de par la propriété découlant de l'eq. (1.6).

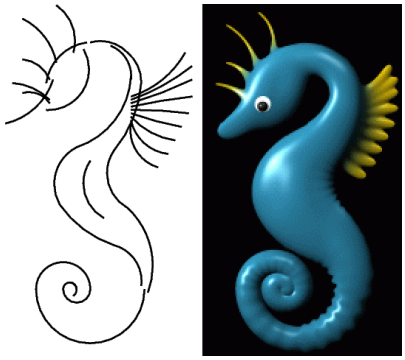


FIGURE 1.12 – *Exemple de surface de convolution générée à partir de squelettes 1D. Tiré de [She99].*

Le principal désavantage de l'utilisation des surfaces de convolutions concerne la nécessité de l'évaluation de l'intégrale de κ pour chaque évaluation du potentiel. De nombreux travaux ont été faits pour développer des fonctions noyaux pouvant s'intégrer analytiquement [She99], cependant, dans le cas général, l'évaluation de celle-ci nécessite le recours à des méthodes d'intégrations numériques basées sur des quadratures [QSS00].

Pour éviter cette évaluation couteuse, il est courant de remplacer l'intégration par une fonction $\min$. On désigne cela par *fonction de distance* généralisées pour des squelettes de dimension non nulles. On peut alors écrire

$$\begin{cases} f_i(\mathbf{p}) = \kappa(\|\mathbf{p}-\mathbf{q}_0\|) \\ \text{avec } \mathbf{q}_0 = \arg\min\limits_{\mathbf{q}\in\mathcal{C}} \|\mathbf{p}-\mathbf{q}\| \,. \end{cases} \tag{1.7}$$

Un calcul de distance minimale [3] étant généralement moins couteux que l'évaluation de l'intégrale.

Bien que répendu, l'utilisation de fonction de distances minimales fait perdre les propriétés des surfaces de convolutions :

1. La fonction de potentiel fait désormais intervenir la fonction non dérivable $\min$. Elle a pour effet que toute discontinuité de dérivabilité du squelette se répercute sur la fonction spatiale dans sa globalité. La surface implicite n'est donc pas lisse.
2. La fonction $\min$ n'est pas linéaire. Il n'y a plus équivalence entre sommation des contributions de plusieurs squelettes, et contribution d'un unique squelette. Il en résulte l'apparition d'un défaut visuel typique de ce type d'approche : un gonflement localisé autour des jonctions entre les différents squelettes (voir fig. 1.13).

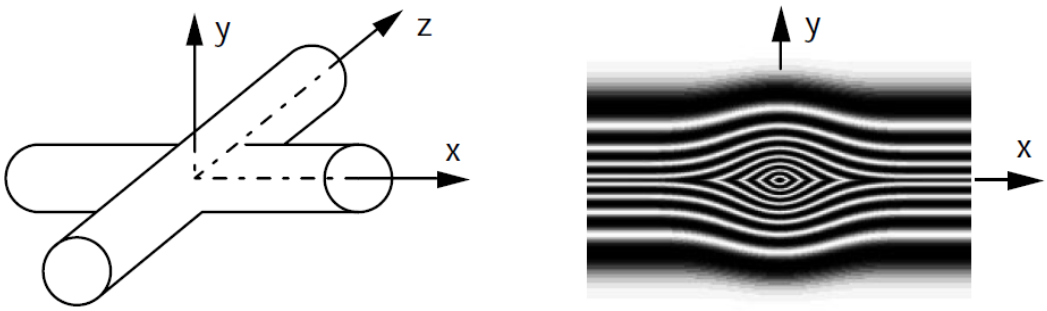


FIGURE 1.13 – *Exemple de gonflement local appelé* bulge *localisé à la jonction entre deux squelettes lors de l'utilisation d'une fonction de distance. La figure de gauche représente les squelettes. Illustration tirée de [Blo97].*

D. Cas du noyau de Cauchy

Nous considérons désormais le cas particulier d'un noyau d'intégration appelé *noyau de Cauchy*[4]. Ce noyau introduit par McCormack et Sherstyuk [MS98] possède l'avantage d'être une fonction lisse rapidement décroissante et dont l'intégrale le long d'un segment est connu analytiquement. On la note κ_c, avec

$$\kappa_c(u) := \frac{1}{(1+s^2u^2)^2}\ , \tag{1.8}$$

avec s un paramètre de contrôle du rayon de décroissance. L'aspect de ce noyau et la primitive implicite associée à un segment est illustré en fig. 1.14.

Il devient alors possible de combiner les avantages des propriétés de mélanges des surfaces de convolution sans avoir à payer le prix d'une intégration numérique couteuse.

Expression de la formule analytique intégrée

Soit un segment $[\mathbf{p}_a, \mathbf{p}_b]$ non dégénéré. $\mathbf{p}$, un point quelconque de l'espace où l'on souhaite connaitre la valeur du potentiel. Soit

$$\begin{cases} L = \|\mathbf{p}_b - \mathbf{p}_a\| \\ d = \|\mathbf{p} - \mathbf{p}_a\| \\ h = \frac{1}{L} < \mathbf{p} - \mathbf{p}_a, \mathbf{p}_b - \mathbf{p}_a > \ , \end{cases}$$

L'expression analytique du noyau de Cauchy intégrée sur le segment est donnée par [MS98] :

$$\begin{aligned} F(\mathbf{p}) = \ & \frac{1}{2p^2}\left(\frac{h}{p^2+s^2h^2} + \frac{L-h}{s^2(L-h)^2+p^2}\right) \\ & + \frac{1}{2sp^3}\left[\operatorname{atan}\left(\frac{sh}{p}\right) + \operatorname{atan}\left(\frac{s(L-h)}{p}\right)\right] \end{aligned} \tag{1.9}$$

3. Les structures de données spatiales de type Octree permettent des évaluations rapides de calculs de distances.

4. Appelé initialement *quasi-Cauchy* par simple ressemblance avec la loi de Cauchy élevée au carré [She99] chap.2.3.1.

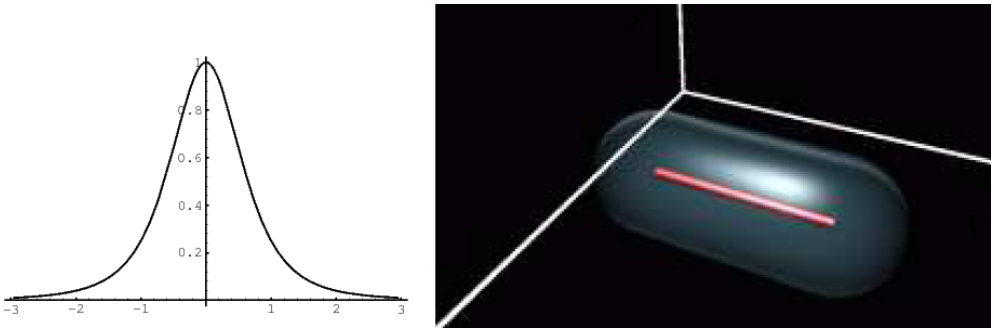


FIGURE 1.14 – *Illustration du noyaux de Cauchy (gauche), et de la primitive implicite (cylindre bleu) associé à un segment jouant le rôle de squelette (rouge). Image tirée de [MS98].*

Cas particulier du plan infini

On notera également le cas particulier de la primitive implicite associé à un squelette 2D formé par un plan infini. Supposons que la position $\mathbf{p}$ soit à une distance d de ce plan, alors le potentiel vaut [MS98] :

$$F(\mathbf{p}) = \frac{\pi}{s^2(1 + s^2\, d^2)} \,. \tag{1.10}$$

Autres modèles discrets implicites

Notre travail utilisera essentiellement les surfaces implicites à squelettes. Il existe cependant d'autres descriptions discrètes possibles. Sans détailler leur description, voici quelques représentations et utilisations classiques :

Les approches variationnelles [TO99] permettant d'interpoler ou d'approximer des positions éparses de l'espace.

Similairement, des approches variationnelles peuvent également être utilisées pour la déformation de surfaces implicites issue de modèles discrets ou paramétriques. En particulier, l'approche dite *level set* [OS88] permet de déformer une surface par application d'un champ de vitesse sur le volume de la fonction. La déformation nécessitant alors la résolution d'un système d'équations aux dérivées partielles sur une grille stockant les échantillons de F. L'application principale de cette approche concerne la modélisation physique de membranes séparant deux phases. Son utilisation dans le domaine graphique [BFM+04] s'est révélée essentielle dans le domaine de la modélisation de fluides [FF01, EMF02] ou de gaz [FSJ01, NFJ02].

Plus récemment, les *Moving Least Squares* (MLS) [ABCO+01, GG07] permettent l'approximation de surfaces implicite à l'aide d'une formulation de moindres carrés locale. Leur application est particulièrement efficace pour la reconstruction de surfaces à partir de nuages de points.

Nous avons présenté deux modes principaux de présentation par surfaces : explicite et implicite. Nous avons vu que chaque type de représentation possédait des avantages et inconvénients propres. Nous verrons que nos contributions s'appuient principalement sur des surfaces représentées par une modélisation polygonale explicite : la plus repandue et la plus générique.

Nous mixerons cependant ces maillages avec une représentation implicite en partie 3.3 afin de tirer avantage des mélanges lisses des fonctions implicites pour la génération de plis de vêtements.

Nous nous intéressons dans la suite de cet état de l'art aux méthodes de déformations, en commençant par les déformations dites *basées physique*.

1.2 DÉFORMATION PAR MODÈLES PHYSIQUES

1.2.1 Principe général

D'une manière générale une déformation physique consiste à exprimer la réponse d'un matériau aux forces à laquelle il est soumis. Si cette réponse s'inspire des lois décrivant un matériau réel, la méthode est dite *physique*.

Les études théoriques et expérimentales décrivant l'évolution d'un matériau sous l'action de forces fournissent dans la grande majorité des cas des lois s'exprimant de manière différentielle, à la fois temporellement et spatialement. L'évolution spatiale et temporelle de la forme s'exprime donc comme solution intégrée d'une équation aux dérivées partielles. L'intégration de ces équations est désignée par *simulation physique*.

Ceci nous permet de préciser les avantages et inconvénients majeurs d'une approche basée *physique*.

Avantages :
- **Réalisme**. La déformation obtenue est visuellement plausible, car elle correspond à une modélisation d'un objet réel.
- **Précision**. Le résultat est quantifiable, au sens où l'erreur consiste à comparer un modèle réel au résultat calculé. Un degré de précision absolue peut donc être défini.
- **Dynamique**. La description est complète. C'est-à-dire que l'ensemble des paramètres de la déformation est pris en compte : sa cinématique, mais également sa dynamique.

Inconvénients :
- **Controle**. Il est généralement difficile de maitriser le résultat final issu de la simulation. En effet, les données d'entrées sont les forces à l'origine de la déformation et non la forme finale souhaitée. Dans le cadre d'une animation de créatures virtuelles, ces forces sont généralement inconnues, et de plus, peu intuitives à paramétrer. Il devient nécessaire de travailler par suite d'essais-erreurs pour tenter de converger vers le résultat souhaité.
- **Coût calcul**. La résolution d'une équation aux dérivées partielles requiert généralement l'utilisation d'outils avancés d'analyse numériques. Ces résolutions numériques sont spécialement couteuses en temps et en mémoire. De plus, certaines contraintes fortes telles que l'absence d'élongation rend les systèmes à résoudre mal conditionnés et génère des risques de divergence numérique.
- **Liberté artistique**. Il n'est pas toujours souhaitable de modéliser des objets *physiquement réalistes*. L'animation expressive dy type de celle utilisée en dessin animé [Las87] nécessiterait par exemple de spécifier de nouvelles lois.

En raison de ces désavantages, les méthodes de déformations que nous proposerons dans nos contributions ne rentreront pas dans la catégorie de la *modélisation physique*. Nous ne réaliserons donc pas un état de l'art détaillé de ces méthodes. Cependant, elles sont d'un intérêt

majeur pour engendrer la dynamique d'un mouvement : ainsi, nous nous baserons sur l'utilisation potentielle de simulations physiques rapides en tant que donnée d'entrée pour l'animation de vêtements dans le chapitre 3.

1.2.2 Déformations physiques standard

Initialement introduite par Terzopoulos *et al.* [TPPF87], les déformations utilisées dans le domaine graphique consistent à modéliser des éléments déformables de type élastique ou plastique.

La relation entre l'effort exercé sur le modèle et la déformation subie est appelée relation constitutive du matériau. Dans le cas d'un matériau continu, la force est mesurée par un tenseur de contrainte σ (stress), et la réponse en déformation spatiale est caractérisée par le tenseur d'allongement relatif (strain) ϵ.
Lorsque ces deux quantités sont proportionnelles, on parle de comportement élastique linéaire. Cette loi est connue sous le nom de *Loi de Hook*. La relation peut alors s'exprimer par

$$\sigma = E\epsilon \,, \tag{1.11}$$

où E est appelé le *module de Young* ou module d'élasticité.

On peut alors montrer que la relation fondamentale de la dynamique liant le déplacement spatiale $\mathbf{u}$ aux contraintes σ et forces extérieurs $\mathbf{F}_{\text{ext}}$ est donnée par la relation différentielle [Spe04]

$$\nabla \cdot \sigma + \mathbf{F}_{\text{ext}} = \eta \frac{\mathrm{d}^2\mathbf{u}}{\mathrm{d}t^2} \,. \tag{1.12}$$

Pour plus de détails sur l'utilisation de ces équations dans le domaine graphique, nous invitons le lecteur à se référer à l'état de l'art de Nealen *et al.* [NMK+05].

Un cas particulier nous intéressant consiste à considérer le modèle virtuel comme étant intrinsèquement discret. Un objet est alors représenté par un ensemble de masses discrètes liées entre elles par des forces agissant similairement à des ressorts. Il s'agit de la représentation la plus simple de modèles physique et est désigné classiquement sous le terme de *masses-ressorts* lorsque la connectivité est prédéfinie, et de *système de particules* lorsqu'elle ne l'est pas.

1.2.3 Cas particulier du modèle à masses-ressorts

Les masses ressorts correspondent au modèle physique le plus simple d'un point de vue théorique et pratique. Ces modèles furent les premiers mis en place sur l'outil informatique [PB81, Mil88].

Rôle et utilisation

De par leur approximation d'un objet continu en positions discrètes, les modèles à ressorts sont peu précis par rapport à leur pendant continu tels que l'approche par volume ou élément finis. Cette faible précision se révèle particulièrement handicapante pour des applications d'ingénierie ou médicales ayant pour vocation de servir de prédiction numérique. Cependant, l'application dans le graphique ne nécessite pas la garantie du caractère prédictif du modèle, mais principalement de garder une apparence visuellement plausible. De ce fait, ce modèle se révèle très adapté dans de nombreux cas d'animation 3D et reste l'une des approches les plus répandues.

Nous décrivons quelques avantages et inconvénients de cette approche, importants pour l'utilisation que nous en ferons :

Avantages

– **Implémentation**. De mise en place simple et intuitive, ce modèle discret s'adapte directement aux modèles polygonaux représentant les surfaces.
– **Rapidité**. La simulation associée est efficace à résoudre. C'est à l'heure actuelle l'une des seules approches [5] permettant d'obtenir des résultats interactifs applicables dans des jeux vidéos.

Défauts

– **Divergence**. La résolution numérique de systèmes à masses ressort peut générer des divergences du système qui se traduisent visuellement par des acoups brutales pouvant aller jusqu'à l'explosion [6]. Pour limiter ce phénomène numérique tout en gardant l'avantage d'une résolution, explicite, il faut considérer des ressorts peu raides, l'ajout d'une dissipation de l'énergie au cours du temps (amortisseurs) et considérer des pas de temps faible. Ceci empêche l'animation rapide d'objets peu élastiques.
– **Contrôle**. La modélisation par masse-ressort ne converge pas vers le résultat physique réel. C'est-à-dire que lorsque le pas d'échantillonnage spatial diminue, le résultat obtenu ne converge pas vers la solution que donnerait un système équivalent réel. Outre le problème de précision numérique, l'inconvénient majeur en graphique revient au fait qu'il n'est ainsi pas possible de prévoir le comportement d'un modèle à haute résolution à partir de tests préliminaires à basse résolution. Un exemple de différence visuelle liée à la subdivision d'un maillage masse-ressort est donné en fig. 1.15.

Modélisation

Dans ce modèle, un objet étant défini par un ensemble de masses discrètes, les seuls degrés de liberté considérés sont ceux de la translation de ces positions. On notera $\mathbf{p}_i$ la position d'une masse discrète i. Ces masses et les caractéristiques qui leur sont associées sont généralement appelées *particules*.
Entre deux particules, le module d'élasticité est considéré comme constant. Il peut ainsi être modélisé par un ressort. La loie de Hook de l'equation 1.11 se résume alors à

$$F_\mathrm{r} = K\ \Delta L\ , \tag{1.13}$$

avec F_r amplitude de la force de rappel du ressort, ΔL l'élongation relative de la particule par rapport à sa position de report, et K scalaire appelée dans ce cas raideur du ressort.

Dans le cas d'un ressort 3D centré en position $\mathbf{p}_0$ de longueur au repos L_0, l'équation vectorielle associée peut s'écrire

$$\mathbf{F}_\mathrm{r} = K\left(\|\mathbf{p}-\mathbf{p}_0\| - L^0\right)\frac{\mathbf{p}-\mathbf{p}_0}{\|\mathbf{p}-\mathbf{p}_0\|}\ . \tag{1.14}$$

Un objet complet est représenté par une structure de graphe contenant un ensemble de sommets V et un ensemble de ressorts R liant les sommets. Potentiellement, chaque sommet de

5. Les modèles à éléments finis avec très peu de composants peuvent cependant être utilisés [PO09] interactivement, de même que certaines représentations moins classiques [GBFP11b, GBFP11a].
6. Positions des particules partant à l'infini.

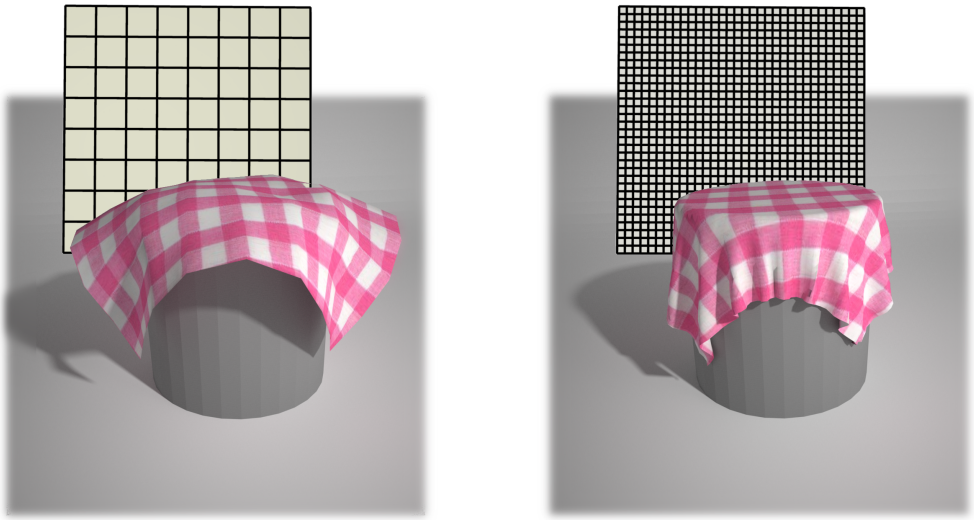


FIGURE 1.15 – *Illustration de la dépendance du comportement d'un modèle masse-ressorts à la finesse d'échantillonnage. Les deux modèles représentent une simulation de tissu sur un maillage dont les arêtes sont considérées comme des ressorts. Les figures du fond représentent les maillages respectifs. La figure de gauche est le résultat de la simulation pour un maillage à basse résolution, la figure de droite est le résultat de la même simulation sur une version subdivisée du maillage. Les paramètres sont définis de manière à simuler des modèles identiques. C'est à dire que les constantes de raideurs par unité de longueur des ressorts ainsi que la masse par unité de surface des particules sont identiques. On observe cependant que le résultat visuel est totalement différent. La simulation est réalisée par le logiciel Blender [Ble].*

V peut être relié à l'ensemble des autres sommets générant alors un grand nombre de ressorts. En pratique, on ne liera les sommets qu'à leurs voisins les plus proches au sens topologique. Ainsi, pour des volumes, on utilise des dessorts diagonaux alors que pour les tissus, on utilise des ressorts vers les voisins d'ordre 2.
Cette structure (V, R) est désignée sous le terme de *réseau*. Elle est suffisamment générique pour permettre la modélisation d'objets volumiques [ChZK98] ou bien surfacique tel que des tissus [BHW94].

Simulation par masse-ressorts

Soit une particule d'indice i, de position $\mathbf{p}_i$ de masse m_i. Les indices des sommets reliés à celle-ci par des ressorts sont désignés par $\mathcal{V}_i$. La raideur d'un ressort liant un sommet i à un sommet j est noté K_{ij}. De même L^0_{ij} désigne la longueur de repos du ressort lié aux sommets i et j.
L'application du principe fondamental de la dynamique d'un objet élastique soumis à la gravité nous donne alors le système d'équations différentielles ordinaire à résoudre :

$$m_i \frac{\mathrm{d}^2\mathbf{p}}{\mathrm{d}t^2} = \sum_{j \in \mathcal{V}_i} \left[K_{ij} \left(\|\mathbf{p}_i - \mathbf{p}_j\| - L^0_{ij} \right) \frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|} \right] + m_i \, \mathbf{g} \, . \tag{1.15}$$

Il s'agit d'un système d'équations non linéaires à N_s inconnues, où N_s représente le nombre de sommets. Sa résolution nécessite l'utilisation de méthodes d'intégrations numériques [QSS00] consistant à faire évoluer les positions itérativement suivant des pas de temps Δt en considérant les forces et une vitesse constante pendant ce laps de temps. Plus les constantes de raideurs K sont grandes, et plus les pas de temps Δt doivent être petits.
Dans le domaine du graphique, il est commode d'utiliser une intégration explicite d'Euler d'ordre 1, ou de Runge-Kutta [HNW00] lorsque l'on modélise des objets élastiques. Dans le cas où K est grand, il est généralement souhaitable de mettre en oeuvre une intégration implicite (Euler implicite) afin de limiter les divergences [BW98]. Celle-ci nécessite la linéarisation de l'équation et la résolution d'un système linéaire à chaque pas de temps [BW98].

1.2.4 Animation physique de tissus

Nous nous intéressons particulièrement à l'animation de tissus et de vêtements portés par des personnages. N'étant pas le point principal de notre approche, nous ne décrirons pas exhaustivement la littérature du domaine. Le lecteur intéressé pourra cependant se référer à des publications dédiées [VT98, KCBZ05, CK05].

Basée sur les mesures physiques de raideurs décrites par Kawabata [Kaw80], les simulations physiques de vêtements ont été étudiées de longue date dans les applications graphiques [CYTT92, BHW94, Pro97, BW98].
Les méthodes de simulations particulaires restent aujourd'hui encore les plus répendues [BMF03, GHF+07] pour des applications interactives bien que la modélisation par la mécanique des milieux continus se développe récemment [EKS03, TPS09].

La simulation de vêtements reposants sur le corps d'un personnage nécessite également la mise en place du traitement de collisions robuste et rapide avec un maillage animé. Ces traitements sortent également du cadre de ce travail. Nous noterons cependant que la méthode de traitement des collisions proposée par Bridson *et al.* [BFA02] est l'une des plus standard. C'est notamment celle-ci qu'utilise le logiciel Blender [Ble] qui nous servira à réaliser nos exemples.

Modélisation

Basé sur les méthodes particulaires, un tissu peut être considéré comme une surface polygonale, généralement quadrangulaire ou triangulaire. Les sommets géométriques de la surface représentent les particules du système masses-ressorts.

Considérons le cas d'une surface quadrangulaire. On distingue différents types de ressorts comme illustrés en fig. 1.16 :

- *Ressorts d'élongations ou structurels* (stiffness). Ils correspondent aux arêtes mêmes du maillage et limitent principalement les élongations entre sommets.
- *Ressorts de cisaillement* (shear). Ils correspondent à des ressorts diagonaux et limitent les glissements des sommets suivant la diagonale des quadrangles.
- *Ressort de courbures* (bending). Ces ressorts sont positionnés par rapport aux voisins d'ordre 2 (2-voisinage) et évitent les variations brutales de courbure de la surface.

Des méthodes plus précises de gestion de la courbure peut également se baser sur le calcul des angles entre faces adjacentes [GHDS03, BMF03], et se généralise au cas des maillages triangulaires.

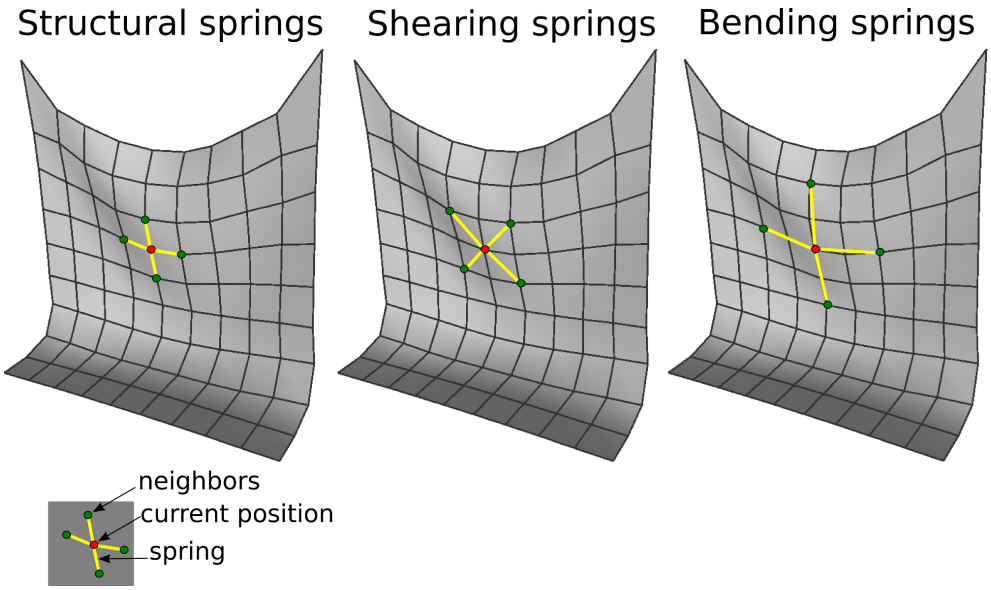


FIGURE 1.16 – *Différents types de ressorts pour une simulation masses-ressorts d'un tissu tombant sous la gravité.*

Problématique

Pour obtenir l'apparence visuelle réaliste d'un vêtement animé, il est essentiel de simuler une surface qui ne s'étire ou se contracte que très peu. En effet, les tissus réels sont formés d'enchevêtrements de fibres [KJM08] (voir fig. 1.17). Celles-ci ne peuvent se déplacer l'une par rapport à l'autre que dans un domaine très restreint correspondant à la taille de la maille. Au-delà de cette distance, la résistance physique des fibres empêche toute autre élongation ou contraction.

Au niveau macroscopique, un tissu peut ainsi être vu comme une surface quasi-inextensible. C'est à dire qu'en tout point de la surface, les longueurs mesurées sur celles-ci restent quasiment préservées. Les vêtements étant fabriqués à partir de patrons planaires, nous pouvons en déduire que la surface d'un tissu doit constamment pouvoir se déplier à plat sur son patron sans élongation ni contraction.

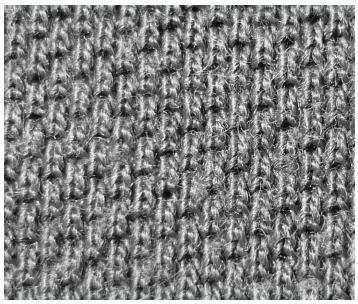


FIGURE 1.17 – *Exemple de mailles d'un tissu.*

Si la surface est suffisamment lisse - au moins C^2 - on parle alors de surfaces développables [Kre91], ou dans le cas d'une préservation approximée des longueurs : quasi-développables.

Au niveau visuel, cette contrainte intrinsèque de la surface est l'origine de la création des plis caractéristiques des tissus. Lorsque des contraintes de directions opposées sont appliquées sur la surface, celle-ci se plisse, mais ne se compresse pas.

Cette propriété intrinsèque de la surface doit donc être traduite par son modèle particulaire physique lors d'une simulation. L'idée la plus simple consiste à augmenter arbitrairement

les constantes de raideur des modèles masses-ressorts standard. Il s'en suit deux problèmes majeurs :

1. **Instabilité**. Vérifiable expérimentalement par la divergence des solutions numériques, Choi et Ko [CK02] montrent qu'une contrainte limitant la compression dans le plan de la surface rend invariablement les systèmes à résoudre mal conditionnés.
2. **Rigidité**. Connu sous le nom de *locking artifact* [Hau04, EB08], l'augmentation de la raideur des ressorts rend le modèle très rigide jusqu'à point de le rendre incapable de se déformer sous l'action de contraintes dans le cas de maillages triangulaires. Hauth [EB08] montre que ce phénomène est lié à un manque de degrés de liberté du modèle.

Corrolaire : Les simulations de type masses-ressorts se révèlent être particulièrement efficaces pour des modèles de surfaces peu rigides à basse résolution. Ils permettent une déformation interactive prenant en compte la dynamique du mouvement tout en gérant les collisions.
Ces modèles ne sont pas contre pas adaptés pour l'animation interactive et contrôlable d'un vêtement inextensible détaillé présentant des plis malgré les différentes améliorations existantes déjà proposées que nous détaillerons en partie 1.4.3.
Notons que Bergou *et al.* [BMWG07] développent une approche permettant d'utiliser une simulation à basse résolution comme guide d'une simulation à haute résolution permettant l'ajout de détails ultérieurs. Cependant, la méthode ne permet pas de gérer les contraintes d'isométries.
Basées sur ces remarques, nous présenterons au chapitre 3 une méthode se basant sur une simulation rapide à basse résolution. L'évitement des compressions par l'ajout de plis se réalise dans une seconde étape purement géométrique, et donc indépendante des problèmes numériques de stabilité.

1.3 DÉFORMATIONS PAR APPROCHES GÉOMÉTRIQUES

Motivation

Dans certain cas, il peut être souhaitable de ne pas suivre une évolution suivant des lois physiques. Le cas classique de personnages de films d'animation aux traits exagérés constitue un exemple d'utilisation de tels procédés comme montré en fig. 1.18.

Les méthodes géométriques se focalisent ainsi non plus sur les lois qui génèrent la déformation, mais sur le résultat de l'animation elle-même. Typiquement, un artiste va vouloir modéliser l'animation qu'il souhaite directement en déformant la forme elle-même. On parle alors de méthode basée résultat ou de modèle descriptif par opposition à modèle générateur.

Ce type d'approche permet d'éviter la complexité de la modélisation devant suivre des contraintes de plausibilités physiques. En effet, modéliser l'animation d'un personnage complet à partir du mouvement physique des os, muscles,



FIGURE 1.18 – *Exemple de personnage virtuel ne suivant pas les lois physiques classiques. Image issue du film d'animation* Incredibles *de* Pixar.

tendons, ...revêt une complexité énorme. De plus, il est relativement intuitif de vouloir déplacer certaines parties du corps manuellement tels que la main, le bras, les jambes, etc. Au contraire, la donnée des forces générant ces actions ne peut pas être effectuée directement par un humain.

Méthodes existantes

Il convient désormais de définir des méthodes de déformation des surfaces que l'on cherche à animer. Dans le cas d'une surface définie par un maillage, il est clair que le déplacement individuel de chaque sommet permet un nombre de degrés de liberté très grand, mais est hors de portée d'un utilisateur. Les outils de déformations géométriques doivent ainsi être un compromis entre les degrés de liberté possibles pour les déformations et la simplicité de spécification de celle-ci. De nombreuses méthodes de déformations géométriques ont déjà été développées et se retrouvent classiquement dans les logiciels de modélisation tels que les déformations par cages, par squelettes, par subdivision, etc, que nous verrons dans ce chapitre.

Nous distinguons dans la suite trois catégories principales :
- **Les déformations de l'espace**, visant à déformer l'espace dans lequel est plongé l'objet.
- **Les déformations basées modèles**, consistant à définir un sous-ensemble dans l'espace des formes possibles à partir d'un faible nombre de paramètres. La forme finale est alors donnée par une combinaison de ces paramètres.
- **Les déformations basées surfaces**, définissant une forme à partir de certaines propriétés que doit vérifier la surface de celle-ci.

1.3.1 Déformations de l'espace

Dans un premier temps, nous nous intéressons aux déformations de l'espace. Ces déformations consistent à considérer l'objet comme étant plongé dans un espace 3D. La déformation n'est alors pas appliquée directement sur la surface elle-même, mais sur l'espace de son plongement. L'avantage de ce type de méthode est qu'il ne dépend pas du type de définition de la forme plongée. Ainsi l'objet peut être défini par un maillage quelconque potentiellement non variété, par une surface implicite [7], ou même simplement par un ensemble de points non connectés.

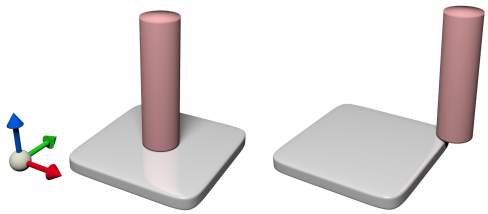
Position du problème : Il est aisé de définir des transformations de l'espace classiques. Les transformations affines telles que la translation, l'agrandissement (scaling), cisaillement (shearing), rotation s'expriment linéairement par une matrice constante (voir fig. 1.19). Si l'on autorise les coefficients des matrices à être des variable en fonction de la position spatiale, on peut également définir certaines transformations telles que les pliures (bending) et vriages (twisting).

Si ces déformations sont suffisantes pour définir localement les déformations les plus courantes, leur définition reste ici globale. Par exemple, si l'on souhaite déformer un bras, on va vouloir limiter l'influence de la déformation à celui-ci et ne pas propager cette déformation au niveau des autres parties du corps. La difficulté majeure consiste alors à rendre leur application locale et leur paramétrage le plus automatique possible. La problématique est de définir un contrôle local sur ces déformations.

7. On définiera alors l'objet déformé par $\tilde{F}(\mathbf{p}) = F(D^{-1}(\mathbf{p}))$, où D est la déformation

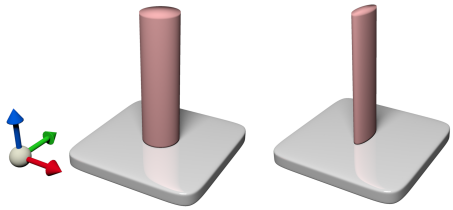
– Translation par (t_x, t_y, t_z)

$$M_{\text{translate}} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



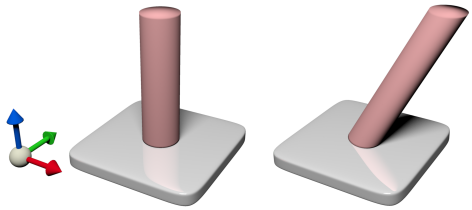
– Aggrandissement par (s_x, s_y, s_z)

$$M_{\text{scale}} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



– Cisaillement

$$M_{\text{shear}} = \begin{pmatrix} 1 & s_{xy} & s_{xz} & 0 \\ s_{yx} & 1 & s_{yz} & 0 \\ s_{zx} & s_{zy} & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



– Rotation d'axe (n_x, n_y, n_z) normé et d'angle φ

$$M_{\text{rotate}} = \begin{pmatrix} \cos(\varphi) + n_x^2(1-\cos(\varphi)) & n_x n_y(1-\cos(\varphi)) - n_z\sin(\varphi) & n_y\sin(\varphi) + n_x n_z(1-\cos(\varphi)) & 0 \\ n_z\sin(\varphi) + n_x n_y(1-\cos(\varphi)) & \cos(\varphi) + n_y^2(1-\cos(\varphi)) & -n_x\sin(\varphi) + n_y n_z(1-\cos(\varphi)) & 0 \\ -n_x\sin(\varphi) + n_x n_z(1-\cos(\varphi)) & n_x\sin(\varphi) + n_y n_z(1-\cos(\varphi)) & \cos(\varphi) + n_z^2(1-\cos(\varphi)) & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

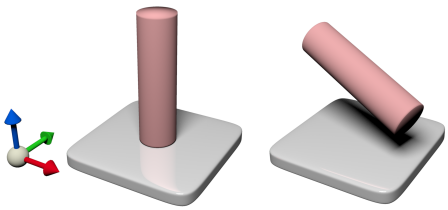


FIGURE 1.19 – *Transformations affines de l'espace.*

Il existe plusieurs approches pour déformer l'espace 3D de manière simple et contrôlable. Dans la suite, nous passons en revue quelques modalités de déformations classiques. Les déformations par *cages*, guidées par le geste ou par des courbes. Nous nous intéresserons ensuite particulièrement aux déformations par squelettes d'animation dont nous nous servirons ultérieurement. Finalement, nous verrons également le principe de la déformation par champ de vecteurs.

En complément de la description qui suit, nous invitons le lecteur intéressé à se référer aux états de l'art publiés sur le sujet [AS06, GB08].

Déformations par cages

Les déformations par cages sont les déformations de l'espace les plus intuitives à comprendre. L'espace 3D est défini par une grille appelé cage répartie dans un domaine de l'espace. En déplaçant les sommets de la cage, on déforme l'espace contenu par celle-ci par interpolation.

Free Form Deformation

Principe Dans le cas le plus simple, la cage est constituée d'une grille cartésienne rectangulaire (lattice) contenant $N_x \times N_y \times N_z$ sommets, orientée suivant les axes globaux de l'espace (x, y, z). Un utilisateur peut alors directement déplacer un ensemble des sommets de la grille. Afin d'obtenir une déformation globalement continue et lisse de l'espace, il est classique d'interpoler l'espace par des fonctions splines. Les positions de la grille servent alors de polygone de contrôle (voir fig. 1.20). Cette approche est connue son le nom de FFD (Free Form Deformation) et fut publiée par [Bar84, SP86, Coq90].

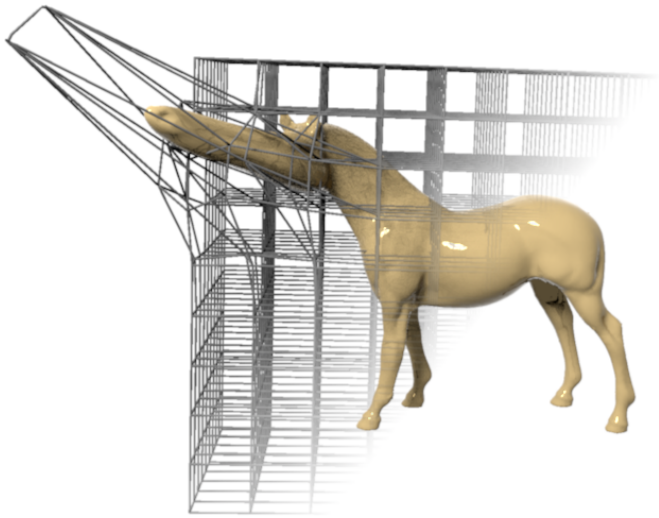


FIGURE 1.20 – *Déformation d'un modèle d'animal par FFD. La surface est plongée dans une grille 3D indiquée par les segments gris. La déformation directe de la grille permet d'appliquer une déformation lisse qui allonge ici le museau de l'animal.*

Théorie Notons $\mathbf{g}_{ijk}$ la position spatiale des points de contrôles de la grille. On peut alors définir la déformation spatiale $\mathbf{f} : (x, y, z) \in [0, 1]^3 \mapsto \mathbf{f}(x, y, z) \in \mathbb{R}^3$ par le produit tensoriel 3D :

$$\mathbf{f}(x, y, z) = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} \sum_{k=0}^{d-1} b_i(x) b_j(y) b_k(z) \mathbf{g}_{ijk} \ ,$$

où $b : u \mapsto b(u)$ sont les polynômes de Bernstein de degré d. Dans la formulation de Bézier, le nombre de points de la grille définit le degré des polynômes.

Classiquement on considérera des déformations Splines de degrés 3 pour une déformation C^2. Le polygone de contrôle étant alors constitué de patchs parallélépipédiques de taille $4 \times$

4×4. Il est aussi d'usage de prendre les B-splines cubiques comme fonctions de base b_i. La discrétisation de la grille est alors décorélée du degré des fonctions de base. De plus, le support local des B-splines fait qu'une modification locale de la grille n'implique qu'une modification locale du modèle plongé dans l'espace.

Limitations Les méthodes FFD sur grille cartésiennes furent l'une des premières utilisées de par leur lien direct avec les fonctions d'interpolation lisses de type spline.

Cependant, l'utilisation de ce type de grille régulière ne permet pas de rendre compte de la topologie d'un objet complexe. En effet, deux objets spatialement proches, mais géodésiquement éloignés suivront inévitablement une déformation quasi identique de par l'interpolation utilisée.
Prenons l'exemple de l'animation d'un personnage placé dans une telle grille. La déformation des sommets de la grille influençant la jambe gauche influencera également la jambe droite. Il n'est pas possible de déformer indépendamment ces deux surfaces spatialement proches.

Cages non structurées

Afin d'offrir un contrôle plus local, il est possible d'utiliser des grilles non structurées.

Des travaux récents proposent ainsi de définir la grille de contrôle par un polygone grossier englobant la surface à déformer (voir fig. 1.21). Si le polygone est suffisamment grossier, il est aisé de déformer celui-ci manuellement. De plus, si celui-ci suit raisonnablement les contours de la forme, la fonction d'interpolation peut approximer la distance géodésique liée à la surface. Ceci permettant de rendre des objets non connexes, mais spatialement proches indépendants.

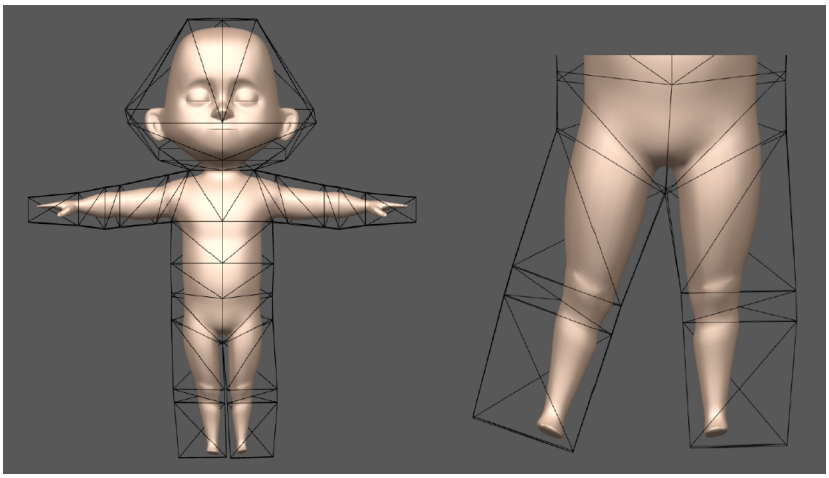


FIGURE 1.21 – *Exemple de déformation par cages. La structure de la cage suit la géométrie du personnage, il devient alors possible de séparer l'influence de la déformation entre les deux jambes du personnage. Image tirée de [JMD+07].*

La cage n'étant plus structurée suivant 3 axes (x, y, z) il n'est plus possible de définir l'interpolation par produit tensoriel de fonctions de bases 1D. Floater *et al.* [FKR05] introduisent une généralisation des coordonnées barycentriques pour un polygone fermé tel que l'on puisse exprimer la position d'un point intérieur de l'espace comme somme convexe de fonctions liées aux sommets du polygone. Lorsque la cage de contrôle est déformée, il devient

alors possible de déformer la surface intérieure par interpolation suivant ces coordonnées barycentriques. Cette idée a été initiée par Ju *et al.* [JSW05], puis implémentée sur GPU par Lipman *et al.* [LKCOL07].

Par la suite, différents types de coordonnées barycentriques ont pu être mises en place afin de respecter au mieux certaines contraintes géométriques lors des déformations telles que le volume, les angles ou les longueurs. On notera notamment le cas des coordonnées harmoniques [JMD+07, BCWG09], biharmoniques [JBPS11] et des coordonnées de Green [LLCO08].

Limitations Les déformations par cages permettent ainsi de déformer un objet quelconque de l'espace en contrôlant explicitement un sous-ensemble de positions de contrôle. Associée à une méthode d'interpolation adaptée, le domaine de déformation peut être défini à l'aide d'une cage polygonale permettant de prendre correctement en compte des parties d'objets spatialement proches, mais non connexes. La construction de la cage reste cependant un travail manuel fastidieu préalable. De même, bien que réduit, le nombre de points de contrôle de la cage reste relativement important pour la déformation d'objets articulés et peut nécessiter des compétences d'animateurs avancés. Enfin, la mise en place de coordonnées d'interpolation non-linéaires reste relativement lourde à calculer à l'heure actuelle et nécessite des accélérations spécifiques afin de pouvoir être applicable en temps raisonnable.

Déformations définies par le geste

Afin d'obtenir une déformation plus aisément manipulable, il est possible de prendre en compte le geste d'un utilisateur (trajectoire + zone d'influence) comme information sur la déformation à réaliser. Pour cela, Angelidis *et al.* [AWC04] définissent une suite de déformations rigides de l'espace le long de la trajectoire définie par l'utilisateur. La pondération de chaque déformation est liée au temps passé sur chaque position. Enfin, les matrices de déformations sont interpolées suivant la méthode d'interpolation matricielle [Ale02]. En notant M la matrice de déformation rigide associée à un instant donné, ϕ la 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

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

avec $\phi \odot \mathrm{M}$ l'opération d'interpolation matricielle $\exp(\phi \log \mathrm{M})$.

Cette méthode a pour but de modeler une forme organique quelconque, comme si elle était en argile ou en pâte à modeler, en partant d'un maillage déjà existant dont le genre topologique ne pourra pas être modifié. Les auteurs mettent en place une méthode interactive dénommée *Sweepers* [AWC04] pour désigner la déformation basée sur le geste temps réel d'un utilisateur.

L'interpolation d'une suite de petites déformations rigides permet de prévenir les auto-intersection. Bien que non démontré exactement sur le plan théorique, le principe de non-intersection est vérifié expérimentalement. Un exemple de telle modélisation est illustré en fig. 1.22.

Déformation par champs de vecteurs

Les déformations précédentes peuvent être vues comme des changements directs de coordonnées donnant la position finale du sommet déformé en une étape. À l'opposé, il est possible

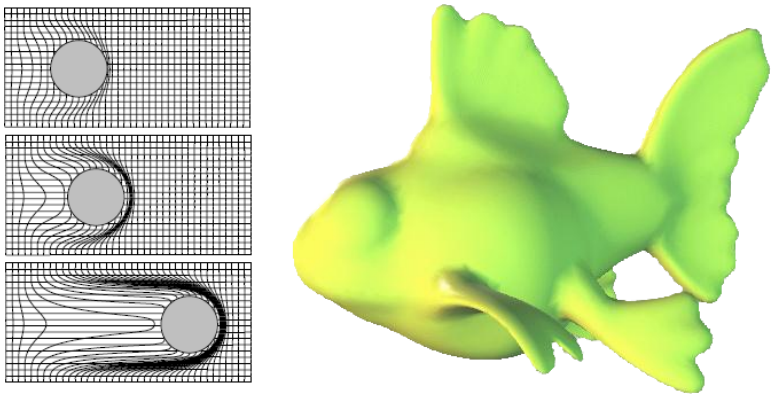


FIGURE 1.22 – *Exemple de modélisation à l'aide de Sweepers. L'image de gauche représente la trace de l'outil dans la grille spatiale qui n'entraine pas d'auto-intersection de l'espace déformé avec lui même, et donc garantit que les maillages déformés restent corrects. L'image est droite est une forme modélisée par cette approche. L'image est tirée de [AWC04].*

de définir une vitesse que doit suivre un sommet à une position spatiale donnée. La position finale étant connue comme intégrale de la position le long du champ de vitesse. Pour cela, on définit dans l'espace un champ de vecteur dont les lignes de champs définissent les trajectoires que vont suivre les sommets de la surface.

En considérant un champ de vecteurs $\mathbf{v} : \mathbf{p} \in \mathbb{R}^3 \mapsto \mathbf{v}(\mathbf{p}) \in \mathbb{R}^3$, la position à l'instant t d'un point de la surface est obtenue par

$$\mathbf{p}(t) = \mathbf{p}_0 + \int_{u=0}^{u=t} \mathbf{v}(\mathbf{p}(u))\, \mathrm{d}u\,, \tag{1.16}$$

avec $\mathbf{p}_0$ position à l'instant de départ.

L'intérêt de cette approche est de pouvoir assurer une déformation satisfaisant certaines contraintes géométriques complexes lorsque l'on considère des champs de vecteurs particuliers.

Von Funck *et al.* [vFTS06, vFTS07] utilisent ainsi ce type de déformation pour la déformation de maillages. Pour cela, l'utilisateur défini spatialement un rayon sphérique d'influence associé à un type de déformation : ex. tirer, tourner, plier, ... Les auteurs définissent procéduralement à chaque type de déformation un champ particulier à l'intérieur de leur rayon d'influence. Au cours de l'application de la déformation par l'utilisateur, les sommets du maillage sont déformés en suivant l'eq. (1.16) avec, au besoin un remaillage et raffinement de la surface lorsque celle-ci se trouve largement étirée ou contractée.

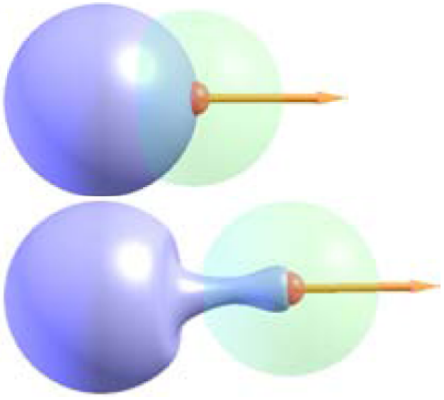


FIGURE 1.23 – *Outil de déformation le long d'un champ de vecteur défini par [vFTS06].*

Limitations Ce type d'approche permet d'effectuer des déformations préservant des contraintes géométriques complexes (nous verrons en partie 1.4.2 leur application dans la modélisation de déformation à volume constant) Son application nécessite cependant l'intégration temporelle des sommets le long d'un champ de vecteur. Les trajectoires suivies ne pouvant se connaitre analytiquement pour les champs utilisées, l'intégration doit se réaliser numériquement et ne permet pas la déformation en temps réel d'un maillage ayant de nombreux sommets. L'implémentation GPU proposée par les auteurs varie entre 1 à 10 secondes par déformation. De plus, le contrôle de ce champ de déformation reste peu intuitif. Il est nécessaire en effet de définir une zone d'influence 3D ainsi qu'un type de champ de vecteurs. Si les auteurs proposent quelques champs particuliers de bases, le type de déformation reste limité, et leurs extensions nécessiteraient la mise en place d'autres champs qui restent difficilement interprétables graphiquement.

Déformation par courbes

Singh et Fiume [SF98] proposent une approche appelée *Wires* qui permet de générer une inflation locale de la surface autour d'une courbe tracée sur celle-ci (voir fig. 1.24). De même, Cutler *et al.* [CGW+05] utilisent une déformation locale d'un maillage autour de courbes animées. Nous nous inspirerons notamment de cette idée au chapitre 3 pour définir des plis de vêtements.

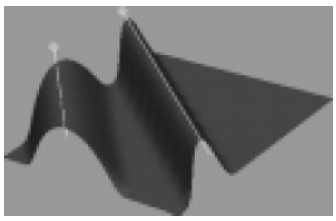


FIGURE 1.24 – *Déformation par courbes. Image provenant de [SF98].*

Ces approches se révèlent intéressantes pour l'ajout de détails sur des surfaces tels que des plis sur des vêtements. Notons que dans le cas de plis de vêtements, il est nécessaire de pouvoir gérer des plis se mélangeant de manière lisse lorsque ces courbes se rapprochent. Notamment, il faut éviter que l'ajout des contributions propre à chaque courbe ne vienne générer un pli de hauteur plus importante au niveau du croisement des courbes. Nous établirons au chapitre 3 une nouvelle approche de déformation par courbe satisfaisant à ces critères.

Déformation par squelettes

Les déformations par squelettes sont des déformations de l'espace [8] contrôlées par la position d'un squelette d'animation : Ensemble hiérarchique de repères (appelé joints) liés par des os (voir fig. 1.25). Ce type d'animation convient principalement aux formes articulées (personnages, mains, ...) de par l'analogie forte entre l'animation d'un squelette d'animation et l'articulation d'un squelette anatomique.

Le principe de l'animation par squelette est le suivant :
- Dans la pose de repos (espace non déformé) chaque partie de l'espace se voit liée à un ou plusieurs repères du squelette.
- L'utilisateur défini une animation du squelette. Pour cela, il est aisé d'articuler manuellement certains os.

8. Bien que ce type de déformation soit généralement appliquée uniquement sur les sommets discrets d'une surface triangulée.

– L'espace se déforme lors de l'animation où la nouvelle position de chaque sommet est calculée comme somme pondérée des positions relatives de ce sommet dans les différents repères en mouvement.

Du fait de ses avantages, nous allons baser l'une de nos approches sur ce type de déformation au chapitre 2. Nous détaillons ainsi avec plus de précision cette approche dans la suite de cet état de l'art. Pour cela, nous allons introduire la déformation par skinning rigide, puis l'interpolation lisse qui en découle. Nous verrons ensuite les problèmes spécifiques liés à cette approche et les solutions déjà proposées. Enfin, nous verrons les principes existants de calculs des poids d'interpolation.

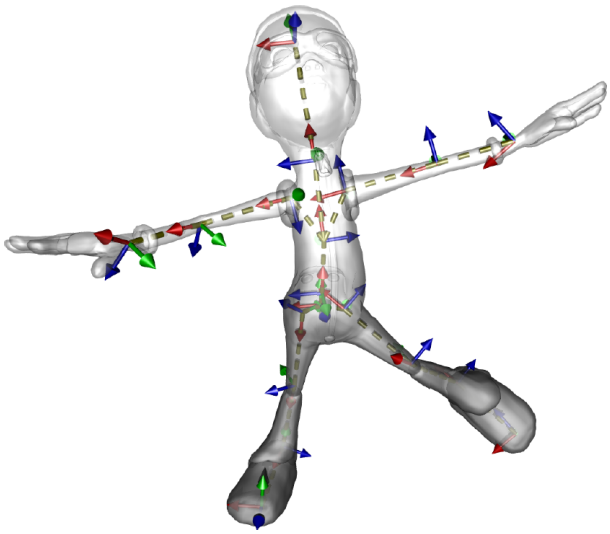


FIGURE 1.25 – *Exemple de squelette d'animation formé par un ensemble de repères hiérarchiques.*

Skinning rigide

Inspiré de la mécanique d'une chaine articulée, le skinning rigide fut l'une des premières méthodes d'animation utilisées pour les jeux vidéos 3D. Pour cela, les sommets du maillage sont attachés rigidement à certains os du squelette [Kro86]. Le nombre d'os total est quelconque, mais chaque sommet ne suit le mouvement que d'un os unique. La surface à animer est ainsi segmentée en sous-partie suivant la déformation d'un joint particulier. L'utilisateur peut ainsi animer certaines articulations du squelette par des translations et rotations, et les groupes de sommets attachés aux os animés suivent ces déformations.

On suppose que $\mathrm{T}_j(t)$ est la matrice de transformation affine associée à au joint j du squelette à l'instant t. On suppose de plus que la position de repos est donnée à l'instant $t = 0$. La position $\mathbf{p}(0)$ d'un sommet quelconque de l'espace peut alors être déformée en suivant la transformation du joint j (on parle de déformation rigide) au cours du temps si elle satisfait la relation

$$\mathbf{p}(t) = \mathrm{T}_j(t)\mathrm{T}_j^{-1}(0)\ \mathbf{p}(0)\ .$$

Cette relation exprime la constance de la position de $\mathbf{x}$ dans le repêre local du joint j. En effet, on a alors pour tout instant t, $\mathrm{T}_j^{-1}(t)\ \mathbf{p}(t) = \mathrm{T}_j^{-1}(0)\ \mathbf{p}(0)$ correspondant aux positions exprimées localement.

La déformation de sous-ensembles de l'espace de manière rigide que l'on désigne par *skinning rigide* n'est cependant pas une transformation globalement continue. En pratique, cela va générer des auto-intersections et élongations majeures de la surface plongée dans cet espace comme illustré en fig. 1.26.

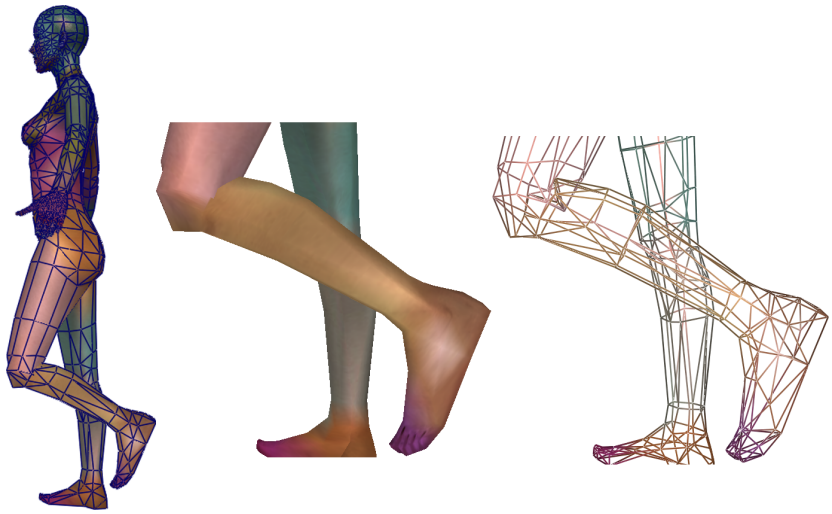


FIGURE 1.26 – *Exemple de skinning rigide de la jambe d'un personnage. Notez l'aut-intersection lors de l'articulation de la jambe.*

Skinning lisse

Principe : Afin de rendre la déformation continue, on permet à une position de l'espace donnée la possibilité de dépendre de plusieurs joints [MTLT88]. L'influence d'un joint envers une position spatiale étant donnée par des poids de pondérations ω également appelé poids de skinning. Si ω est une fonction continue de l'espace, alors la déformation globale le sera également (voir fig. 1.27).

Mise en équations : L'implémentation la plus simple d'une telle pondération consiste à utiliser une interpolation linéaire des matrices de transformations. On appelle cette déformation : LBS (Linear Blend Skinning). En dénotant par N_j le nombre de joints du squelette

d'animation, on a alors :

$$\begin{cases} \mathbf{p}(t) = \left(\sum_{i=1}^{N_j} \omega_j(\mathbf{x}(0)) \ \mathrm{T}_i(t)\mathrm{T}_i^{-1}(0) \right) \mathbf{p}(0) \\ \forall \mathbf{p} \in \mathbb{R}^3, \forall i \in [\![1, N_j]\!] \,, \ \ 0 < \omega_i(\mathbf{p}) < 1 \\ \forall \mathbf{p} \in \mathbb{R}^3 \,, \ \ \sum_{i=1}^{N_j} \omega_i(\mathbf{p}) = 1 \,. \end{cases} \tag{1.17}$$

Les deux dernières conditions concernent l'utilisation d'une combinaison convexe (barycentrique) des pondérations afin déviter l'extrapolation des transformations.

Avantages et inconvénients La déformation par skinning lisse est très répandue dans le domaine graphique. Elle est à ce titre référencée sous divers noms [MTLT88, Lan98, Lan99] : *smooth skinning*, *Skeletal Subspace Deformation* (SSD), *Linear Blend Skinning* (LBS). Nous pouvons recenser quelques avantages majeurs.

- **Efficacité :** C'est une déformation standard des logiciels de modélisation et se retrouve directement utilisé à la volée dans de nombreux jeux vidéos. Cet attrait est notamment du à l'efficacité de son implémentation consistant essentiellement en une somme et multiplication matricielle comme l'exprime la première relation de l'eq. (1.17).
- **Intuitif :** La manipulation directe des os sous-jacents à une surface est très intuitive pour déformer un personnage articulé.
- **Controle :** La modification des poids de skinning permet de faire varier l'apparence de la déformation. Une variation rapide des poids de skinning mimera une articulation franche, alors qu'une variation douce donnera l'apparence d'une courbure lisse.

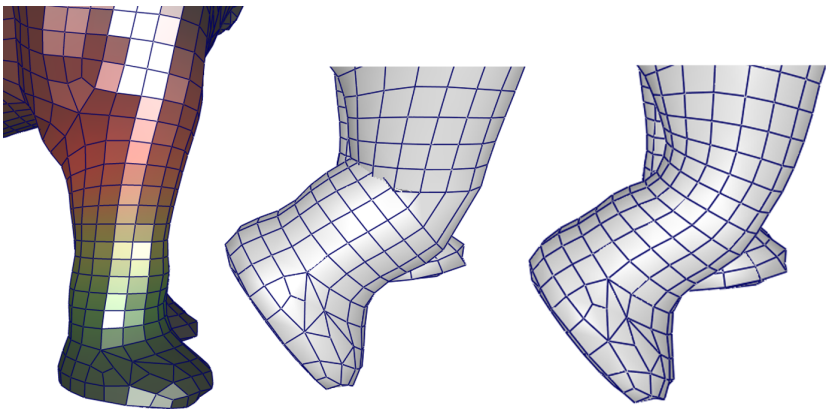


FIGURE 1.27 – *Comparaison entre skinning rigide et lisse. L'image de gauche représente les poids d'interpolations pour un skinning lisse sur la patte d'un modèle d'éléphant. La figure de droite représente l'articulation de cette patte suivant ces poids, à comparer avec le résultat obtenu par un skinning rigide.*

Malgré son utilisation massive (disponible dans les logiciels standards Maya [May], Blender [Ble], ...), la méthode de déformation par skinning souffre cependant d'un défaut majeur lors son utilisation pour des rotations importantes. L'espace interpolé entre la rotation de deux joints est compressé. Ce phénomène engendre une perte de volume local visible lors de son utilisation sur des surfaces définissant un domaine clos tel qu'un personnage. Ce phénomène classique est connu sous le nom de *collapsing-elbow effect* de par visibilité au niveau du mouvement du coude d'un personnage (voir fig. 1.28).

Afin de limiter cet effet, les artistes définissent les poids de skinning de manière obtimale en introduisant des interpolations plus franches au niveau des articulations. La définition de ce type de poids doit cependant ce réaliser manuellement et demande une grande expertise. En effet, bien que les logiciels de modélisation permettent de *peindre* directement sur la surface les influences respectives des joints vis à vis des sommets, les contraintes de normalisation des poids due à la combinaison barycentrique rend leur manipulation peu intuitive.



FIGURE 1.28 – *Illustration du collapsing elbow : perte de volume au niveau du coude.*

De plus, certaines configurations de déformations ne peuvent pas être modélisées par cette approche. Dans le cas de la rotation d'un bras autour de son axe (*twist*), l'interpolation de deux matrices dont les angles sont pivoté par un angle de π radians entrainera invariablement une somme matricielle moyenne nulle au niveau de l'articulation. Ce phénomène est désigné par *candy-wrapper effect* pour sa ressemblance à l'apparence d'un papier de bonbons. Afin de traiter ce problème directement, il est nécessaire de modifier l'algorithme d'interpolation.

Origine du problème : Le problème de la perte de volume provient de l'interpolation utilisée par la méthode du LBS (voir fig. 1.29). En effet, l'interpolation linéaire de matrices de rotations n'est pas une opération stable dans l'espace des rotations. Ainsi la matrice résultante de l'interpolation barycentrique de la première ligne de l'eq. (1.17) n'est plus forcément une rotation, bien que les matrices T le soient.

Plusieurs approches ont déjà été proposées dans la littérature afin de pallier à ce problème d'interpolation linéaire. On peut distinguer deux grandes catégories d'approches :

- Celles basées sur l'apprentissage visant à permettre une déformation proche de celles fournies en exemple comme modèles de références.
- Celles basées sur des modes d'interpolations plus généraux visant à corriger intrinsèquement les défauts de l'interpolation linéaire.

Solutions proposées

Nous détaillons ci-après différentes propositions d'améliorations du skinning suivant les approches principales : Apprentissage, et interpolation.

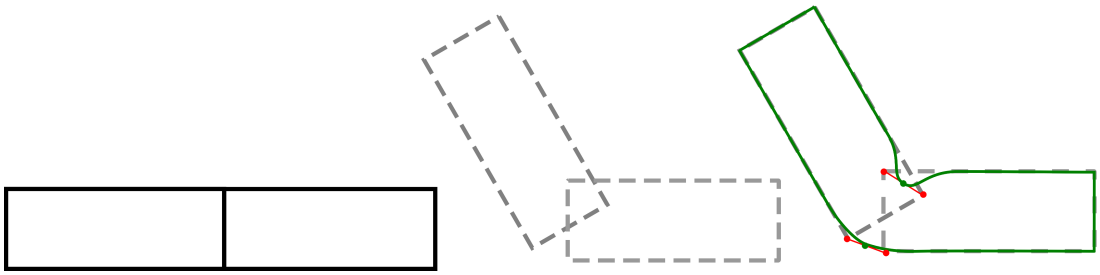


FIGURE 1.29 – *Origine de la perte de volume locale illustrée sur l'articulation de deux tubes vue en coupe. La figure de gauche représente la forme au repos. La figure du milieu est celle du skinning rigide après articulation d'un cylindre. La figure de droite est celle obtenue par skinning rigide. L'interpolation linéaire entre les sommets situés dans leur base locale illustrée par le segment rouge impose la perte de volume dans l'intérieur du coude.*

Méthodes par apprentissage

i **Méthodes basées exemples.**
Lewis *et al.* [LCF00] proposent une amélioration du skinning sous la dénomination de Pose Space. Pour cela, des formes clés de référence sont définies manuellement par un artiste pour un certain nombre de poses du squelette. Ensuite, lors de la déformation, un vecteur de correction est ajouté aux coordonnées des sommets skinnés de manière à minimiser l'écart aux formes de références. En notant $\mathbf{p}_s$ les coordonnées du sommet skinné et δ le vecteur de correction, la position finale du sommet $\mathbf{p}$ est donnée par $\mathbf{p} = \mathbf{p}_s + \delta$ (voir fig. 1.30). Le vecteur de correction étant alors obtenu par interpolation dans l'espace des formes clés.

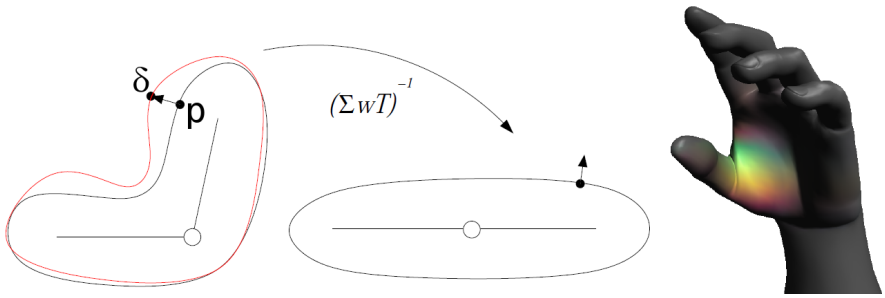


FIGURE 1.30 – *Exemple de correction basée exemples pour la déformation d'une main skinnée dans [KJP02].*

Similairement, Sloan *et al.* [SRC01] réalise également une correction par poses clés, mais cette fois, les formes clés sont stockées dans l'espace des formes de repos modifiées, avant la déformation par skinning.
L'approche nécessite cependant de stocker les vecteurs de déformations δ pour chaque sommet et chaque position clé ce qui rend son utilisation couteuse en mémoire lorsque de nombreuses poses sont définies. Pour limiter ce coût, Kry *et al.* [KJP02] extraient les composantes principales de la correction par analyse spectrale en pre-processing et ne garde que les composantes les plus signifiantes. Lors de la déformation, la correction δ

est obtenue par la somme pondérée des composantes principales. Pour obtenir une déformation interactive, les auteurs proposent pour cela l'implémentation de cette somme sur GPU.

ii **Ajout de joints.**
Morh et Gleicher [MG03] proposent d'ajouter des os au squelette d'animation de manière automatique afin de limiter l'angle de rotation associé à chaque joint. Les os à ajouter ainsi que leur emplacement sont définis de manière à minimiser l'écart entre la surface ainsi skinnée et des poses de références définies manuellement au préalable.

iii **Matrices d'influences**
Wang et Phillips [WP02] étendent les poids de skinning à des matrices d'influence. L'utilisation de matrices permet d'accroitre le nombre de degrés de liberté et ainsi de pallier aux limitations classique du LBS.
En utilisant les notations de l'eq. (1.17) la nouvelle déformation peut alors s'exprimer par

$$\mathbf{p}(t) = \sum_j \Omega_j(\mathbf{p}(0))\, \mathrm{T}_j(t)\mathrm{T}_j^{-1}(0)\, \mathbf{p}(0)\ ,$$

où $\Omega_j(\mathbf{p})$ représente une matrice 4×4 des poids d'influences du joint j.
Le choix des valeurs des matrices d'influences est cependant non intuitif. Impossibles à définir manuellement, ces matrices nécessitent un fitting de leurs paramètres réalisé à partir des formes de références.

Méthodes géométriques

i **Déformations procédurales**
Yang et Zang [YZ05] proposent une déformation de skinning modifiée dans le cas de surfaces tubulaires guidées par une unique articulation. Pour cela, les auteurs définissent des poids d'interpolation ainsi qu'une déformation supplémentaire des sommets comme une fonction de l'angle de pliure. La seconde déformation des sommets permettant de minimiser l'effet de collapsing ellbow et mimant l'apparence de plis de chaires. En dénommant $\theta(t)$ l'angle de l'articulation des os adjacents à l'instant t, le skinning modifié s'exprime par

$$\mathbf{p}(t) = \sum_j \alpha_j(\theta(t))\, \mathrm{T}_j(t)\mathrm{T}_j^{-1}(0)\, \mathbf{p}(0) + \delta(\theta(t)),$$

avec δ la seconde déformation paramétrée d'après l'angle θ évitant l'effet de *collapsing ellbow* et modélisant l'effet de bourrelet.
La méthode se basant entièrement sur une paramétrisation de l'angle de pliure du squelette, elle ne s'adapte pas à une déformation quelconque d'une surface non cylindrique liée à trois os ou plus.

ii **Interpolations matricielles**
Plutôt que de se restreindre à des cas particuliers, certains auteurs proposent directement une interpolation adaptée des matrices de déformations. En effet, l'interpolation linéaire de plusieurs matrices de rotations n'aboutit pas à une matrice rotation. L'utilisation d'opérateurs de déformations stables dans l'espace des rotations permet ainsi d'espérer limiter les défauts classiques du LBS.

Magnenat-Thalmann *et al.* [MTCSP04] utilisent une interpolation définie par exponentielles de matrices introduites dans le domaine du graphique par Alexa [Ale02] (voir fig. 1.31). Dans ce cas, la déformation par skinning s'exprime par

$$\mathbf{p}(t) = \bigoplus_j \omega_j(\mathbf{p}(0)) \odot \mathrm{T}_j(t)\mathrm{T}_j^{-1}\,\mathbf{x}(t)\,,$$

avec $\bigoplus_j \omega_j \odot \mathrm{T}_j = \exp\left(\sum_j \omega_j \log \mathrm{T}_j\right)$.

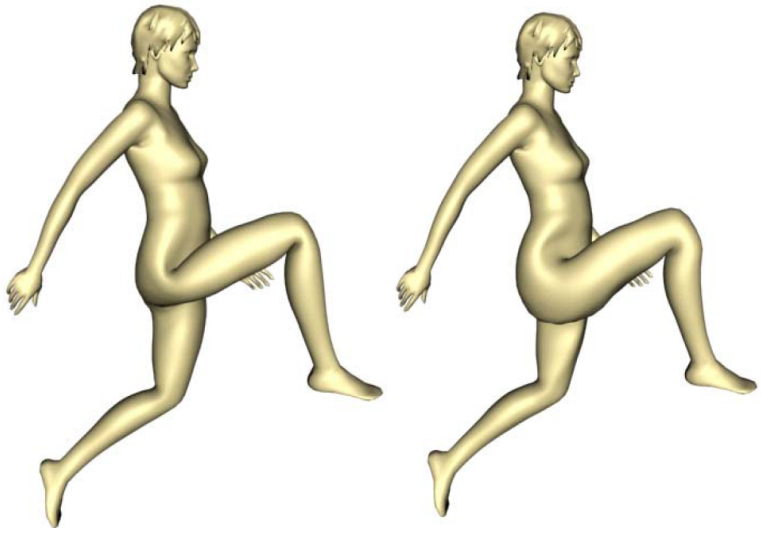


FIGURE 1.31 – *Interpolation linéaire (gauche) comparée à une interpolation par exponentielle de matrices (droite). Image provenant de [MTCSP04].*

Cette approche possède l'avantage d'être suffisamment générale pour traiter l'interpolation de matrices quelconques (rotation, translation, agrandissement, etc). Cependant le calcul de l'exponentielle des matrices nécessite un calcul itératif numérique couteux ne permettant plus l'utilisation du skinning en temps réel.
Afin d'obtenir une interpolation moins couteuse à calculer, Kavan et Zára [KZ05] considèrent le cas particulier des rotations et utilisent une interpolation dans l'espace des quaternions. Cette approche est désignée par *Spherical Blend Skinning*.

Deux difficultés sont discutées par les auteurs :

1. Premièrement, il faut réaliser l'interpolation de plus de deux quaternions pour certains sommets. Le classique *Spherical LinEar intERPolation* (SLERP) ne peux être utilisé dans cas. Les auteurs préconisent l'utilisation d'un moyennenage linéaire des composantes des quaternions. Bien que les propriétés de parcours à vitesse constante du SLERP ne sont pas préservées par cette approche, la méthode garde l'avantage d'être close dans l'espace des rotations contrairement à l'interpolation linéaire des matrices.
2. Deuxièmement, il est nécessaire de définir le centre de rotation à interpoler, car celui-ci doit être considéré indépendamment de la représentation par quaternion. Pour cela, les auteurs calculent ainsi automatiquement un centre de rotation *optimal* minimisant la part de translation liée à la déformation.

Lorsque plus de deux os influencent un sommet, la minimisation est basée sur une décomposition de matrices en valeur singulières. Les auteurs notent que cette approche ne permet plus une déformation en temps réel. Pour cela, ils proposent de stocker en pré-calcul la position du centre de rotation pour différentes orientations des joints.

Au final, l'approche du Spherical Blend Skinning permet une mise en oeuvre plus rapide que l'interpolation basée sur l'exponentielle de matrices. Cependant, le traitement particulier des centres de rotations fait perdre une partie de l'avantage de l'utilisation des quaternions. Quelques années plus tard, les mêmes auteurs proposent alors une méthode unifiée traitant directement les rotations et la partie translation.

iii **Quaternions duaux**

Les quaternions duaux (*dual quaternion*) sont une représentation généralisée des quaternions prenant en compte les rotations et les translations de manière duale. Cette représentation connue depuis longtemps dans le domaine de la mécanique a été introduite dans le domaine graphique pour le skinning par Kavan *et al.* [KCZO07, KCZO08] et permet de formaliser élégamment l'interpolation de rotations ayant des centres de rotations différents (voir fig. 1.32).



FIGURE 1.32 – *Résultats obtenus après application d'une déformation par quaternions duaux (droite) par rapport à l'interpolation linéaire (gauche). Image issue de [KCZO07].*

Notons q_r le quaternion associé à la partie rotation de la transformation, et ε le nombre dual défini par $\varepsilon^2 = 0$. Le quaternion dual associé à la translation par le vecteur (t_x, t_y, t_z) s'exprime par $q_t = (\varepsilon\, t_x/2, \varepsilon\, t_y/2, \varepsilon\, t_z/2, 1)$. Finalement, le quaternion dual exprimant la rotation et la translation est donné par

$$q = q_t q_r \ .$$

En dénotant q_j le quaternion dual associé à la déformation du joint j, le skinning par quaternion duaux peut s'exprimer par

$$\mathbf{p}(t) = \frac{\sum_j \omega(\mathbf{p}(0))\ q_j q_j^{-1}}{\left\| \sum_j \omega(\mathbf{p}(0))\ q_j q_j^{-1} \right\|} \mathbf{p}(0) \ .$$

On notera que tout comme le cas des quaternions classique, l'interpolation suivant une courbe de plus court chemin pour plus de 2 transformations n'est pas exprimable analytiquement. Les auteurs utilisent donc une moyenne normalisée dans l'espace des quaternions duaux.
L'interpolation par quaternions duaux propose donc un formalisme général pouvant être calculé à vitesse interactive. De plus, des implémentations sur carte graphique sont disponibles aisément. Cette approche est désormais devenue commune dans les logiciels de modélisations tels que Maya ou Blender qui proposent l'interpolation par quaternions duaux en standard.
Notons cependant que l'interpolation par quaternions duaux, bien que présentant moins de défauts apparents que l'interpolation linéaire, ne définit pas une déformation à volume constant. De plus, on pourra remarquer expérimentalement que le résultat obtenu est très sensible aux choix des poids de skinning dont nous verrons dans la partie suivante que leur définition reste parfois délicate.

La méthode de skinning a ainsi été largement étudiée de par son utilisation massive en production pour l'animation de personnages. Nous pourrons également noter que récemment, des méthodes de recherche proposent son utilisation dans le cas plus général de déformations d'objets non articulés. Si l'on considère un squelette d'animation formé d'un ensemble de repères qui ne sont plus organisés hiérarchiquement, on peut ainsi définir une déformation de l'espace pour des objets quelconques [JT05, GBFP11a, GBFP11b]. En particulier, le cas de l'animation de tissus est modélisable par ce type d'approche [KMD+07, KSO10] (voir fig. 1.33).

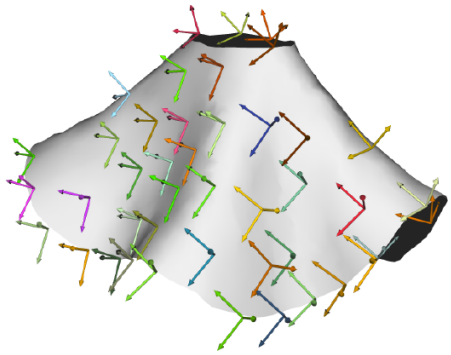


FIGURE 1.33 – *Exemple de modèle de vêtement déformé par repères non hiérarchiques dans [KMD+07].*

Calcul des poids de skinning

La déformation par skinning fait intervenir les poids d'interpolations α qu'il est nécessaire de définir. Comme vues précédemment, des méthodes d'apprentissages peuvent permettre de définir ceux-ci automatiquement.

Cependant, l'utilisation du skinning standard sur un modèle unique est l'approche la plus utilisée. Il est ainsi nécessaire de définir manuellement ou de manière géométrique les poids scalaires d'interpolation.

Définition manuelle

La méthode la plus générique disponible dans l'ensemble des outils de modélisation consiste à peindre directement sur la surface les poids de skinning. Certains artistes numériques dits *skinneurs* possèdent une grande expertise pour ce type de manipulation.

Cette approche laisse toute liberté artistique. Cependant, la normalisation des poids imposée par l'interpolation convexe rend cette manipulation peu intuitive. En effet, la modification

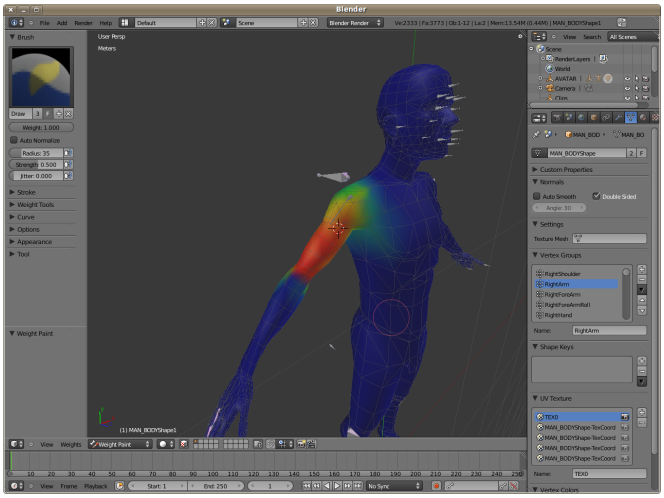


FIGURE 1.34 – *Interface de peinture des poids de skinning associé à l'os du bras (logiciel Blender [Ble]).*

d'un poids pour un os donné vient directement impacter la répartition pour les autres os attachés à ce sommet. Au final, il est difficile de gérer une répartition homogène avec des transitions douces sur l'ensemble de la surface.

Si quelques modifications locales manuelles s'avèrent souvent indispensables, un calcul automatique de poids de skinning pouvant servir de base permet d'éviter une étape fastidieuse de définition sur l'ensemble de la surface.

Distance au squelette

Une approche classique de définition automatique de poids de skinning consiste à utiliser la distance cartésienne au squelette d'animation. Soit un sommet donné et un os j. Soit d_j la distance cartésienne la plus courte entre le sommet et le segment correspondant à l'os j. Les poids de skinning en ce sommet sont alors définis par

$$\alpha_j = \left(\frac{1/d_j}{\sum_{k=1}^{N_j} 1/d_k} \right)^a , \tag{1.18}$$

avec a un paramètre modifiant la vitesse de transition d'influence entre les os généralement choisis entre 1 et 10.

Cette approche se révèle être un bon point de départ pour des poses aux repos (ou *bind poses*) dites *étoilés* comme montrées en fig. 1.35 (gauche). C'est-à-dire que la distance cartésienne de la surface aux os reflète réellement l'influence de ceux-ci.

Cette approche est cependant mal adaptée lorsque certains os proche spatialement vont venir influencer des parties topologiquement éloignées. Typiquement l'utilisation d'une telle méthode pour un personnage dont les deux jambes sont proches l'une de l'autre va créer une dépendance non souhaitée entre l'articulation de la jambe droite et les sommets de la jambe gauche, ou de même, les doigts d'une main risquent de s'influencer les uns les autres comme montré en fig. 1.35 (droite).

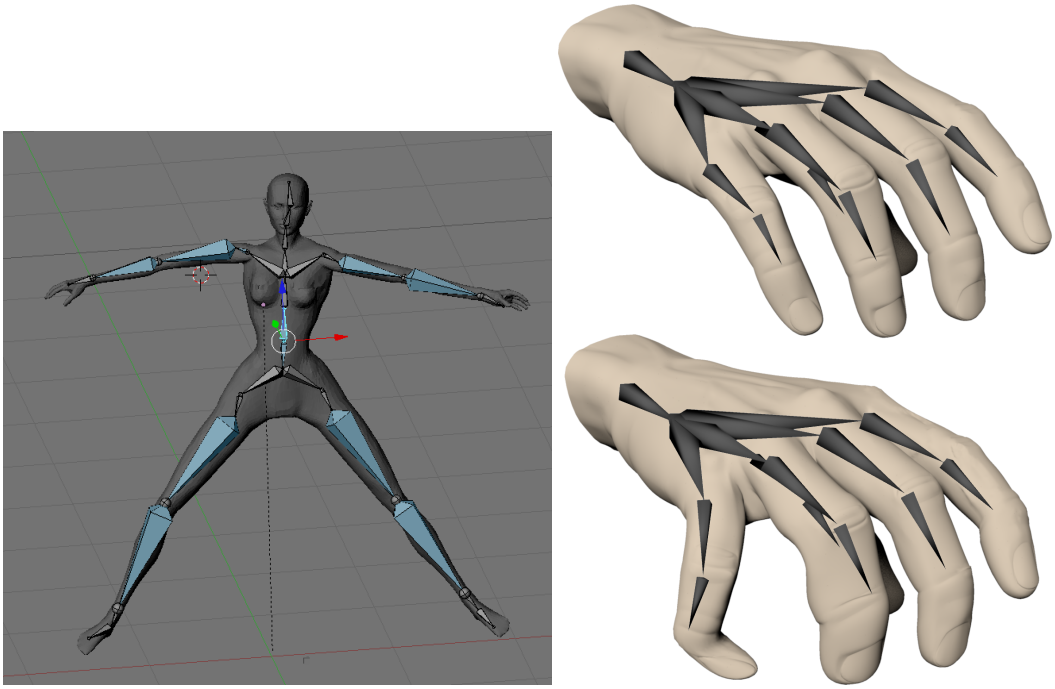


FIGURE 1.35 – *Gauche : Personnage dans une pose de repos dont l'influence des os est directement corrélée à la distance cartésienne aux sommets. Droite : Main déformée par skinning suivant des poids de skinning non appropriées. L'extremité de l'index est lié à l'os du majeure. Son influence empêche les sommets de l'index de suivre un mouvement naturel.*

Distance modifiée Afin de limiter cette influence entre os et sommets non corrélés bien que la distance cartésienne soit proche, Baran et Popovic [BP07] proposent l'utilisation d'une distance cartésienne modifiée. L'approche se déroule en deux étapes :

1. Considérons le rayon joignant le sommet courant au point le plus proche de l'os j. Si ce rayon est entièrement contenu dans la forme, c'est à dire n'ayant pas d'intersection avec la surface, alors $d_j^+ = d^j$. Dans le cas contraire, on prend $d_j^+ = +\infty$, c'est-à-dire que l'influence de cet os sera nulle.

   Finalement, on affecte les poids en utilisant l'eq. (1.18) en considérant cette fois une distance modifiée donnée par d^+.

2. On lisse la distribution des poids en résolvant une équation de diffusion sur la surface. En considérant α_j la distribution de poids sur la surface, on résout de manière discrète l'équation

$$\frac{\partial \alpha_j}{\partial t} = \triangle \alpha_j + \frac{A}{d_j^+}(c_j - \alpha_j)\,,$$

   avec le terme similaire à une source de chaleur : $A \simeq 0.22$ permettant de s'approcher d'une diffusion volumique d'après les auteurs, et c_j valant 1 lorsque l'os le plus proche suivant la distance d^+ est l'os j, et $c_j = 0$ sinon.

   Cela permet de diffuser l'influence des poids de skinning comme montré en fig. 1.36 tout en considérant une propagation géodésique tenant compte de la forme de la surface.

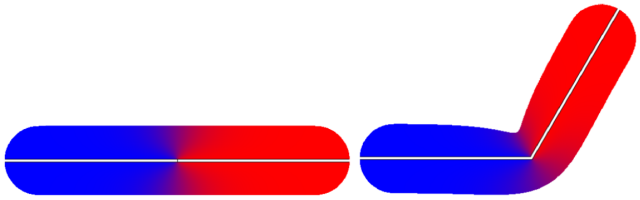


FIGURE 1.36 – *Exemple de diffusion de poids de skinning par équation de la chaleur sur un cylindre. Illustration issue de [BP07].*

Cette méthode permet l'obtention d'une bonne approximation dans le cas de surface simple. La distance volumique modifiée permet de tenir compte de la géométrie de la surface par l'utilisation de calculs d'intersections. Le lissage par diffusion permet quant à elle la prise en compte d'une distance géodésique surfacique afin d'assurer des transitions douces sans introduire d'influences non désirées.

On pourra également noter que Weber *et al.* [WSLG07] utilisent une approche similaire en réalisant l'étape 1 à partir d'un skinning rigide uniquement. Waregham *et al.* [WL08] proposent une variante basée sur un calcul de visibilité pour éviter une diffusion trop large lors de l'étape 2. Enfin Hetroy *et al.* [HGLT09] mettent en place une méthode de calcul de poids automatique s'appliquant aux surfaces formées par des assemblages cylindriques.

Nous pouvons cependant noter que le calcul d'intersection par lancé de rayons possède un coût non négligeable si le temps de calcul est crucial. Nous proposerons ainsi en partie 2.1 une approximation rapide de cette approche dans le cas de géométrie localement convexe.
Également, dans le cas où la géométrie de la surface est relativement complexe, c'est à dire possédant de larges concavités tel que des replis de peau, l'étape 1. de la méthode décrite ne permet pas une initialisation correcte de ces poids. Nous proposerons ainsi une extension par un calcul de distance géodésique robuste pouvant gérer de telle géométrie de manière automatique également en partie 2.1.

1.3.2 Déformations par l'exemple

Nous décrivons dans cette partie quelques approches classiques des déformations basées modèles. Ces déformations consistent à paramétrer un ensemble de formes possibles à l'aide d'un faible de nombre de paramètres. En faisant varier la combinaison de ces paramètres, il est alors possible de décrire cet espace de formes.

On pourra distinguer deux approches possibles :

1. L'interpolation directe de formes de bases.
2. L'extraction d'informations par apprentissage sur un ensemble de formes.

Ces deux approches possèdent l'avantage de décrire des déformations arbitrairement complexes avec peu de paramètres. La difficulté principale étant de définir ces paramètres de manière optimale pour permettre à la fois de décrire un ensemble suffisamment grand de déformations tout en gardant un contrôle intuitif. Nous décrivons dans la suite les principes fondamentaux de ces approches et quelques applications proches des déformations que l'on souhaite modéliser. Cependant, nous ne réaliserons pas une étude exhaustive des publications de ce large domaine.

Interpolation de formes

La méthode principale des déformations basées modèles est le principe de l'interpolation de formes clés, connu également sous le nom de *blend-shape* ou simplement *Morphing*.

L'idée initiale consiste à pré-définir un ensemble de formes de bases appelées formes clés. Appelons $(S_i)_{i\in[\![1,N_p]\!]}$ l'ensemble de ces N_p formes. Une pose déformée S est alors définie comme l'interpolation dans cet espace de formes de dimension N_p

$$S = \sum_{k=1}^{N_p} \psi_k S_k \ ,$$

où les poids scalaires d'interpolations ψ_k réalisent une interpolation convexe

$$\begin{cases} \forall k \in [\![1, N_p]\!] \ , \quad \psi_k > 0 \\ \sum_{k=1}^{N_p} \psi_k = 1 \end{cases}$$

Ce type de déformation est largement utilisée pour l'animation faciale [Par72, LA10] comme illustrée en fig. 1.37. En effet, les mouvements d'une face humaine sont à la fois complexes à modéliser, et de plus nous sommes très sensibles à l'intonation donnée. Il est ainsi plus aisé de naviguer entre des formes connues et validées à l'avance plutôt que de les re-créer à partir d'outils géométriques.

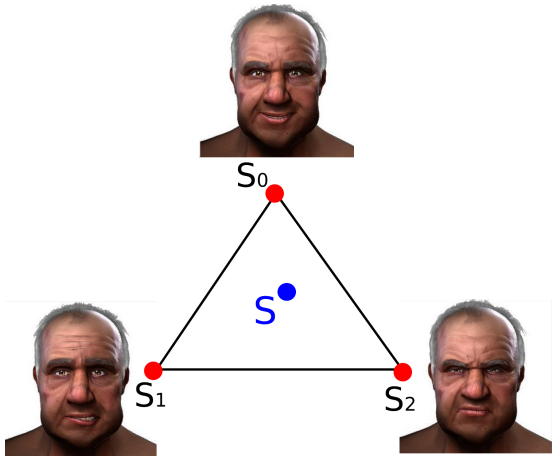


FIGURE 1.37 – *Exemple d'interpolation faciale entre 3 poses clés. Images tirées d'exemples du logiciel Maya [May].*

Bien que triviale à mettre en oeuvre et rapide à calculer, l'utilisation de l'interpolation linéaire (ou polynomiale) ne préserve pas les caractéristiques des formes au cours de l'interpolation, en particulier elle est très dépendante à l'orientation de celles-ci. Il en résulte l'utilisation d'un grand nombre de formes intermédiaires modélisées manuellement. Par exemple, le personnage de Gollum du film d'animation *Le Seigneur des Anneaux, Les Deux Tours* a nécessité la modélisation et l'utilisation de 675 formes clés [For03]. Des méthodes d'interpolations plus avancées [SRC01, KMP07, WDAH10] permettent l'utilisation d'un nombre plus restreint de poses, tout en préservant l'apparence plausible de la forme au cours de la déformation et s'appliquant ainsi à l'animation de personnages articulés. Certaines permettent également l'extrapolation de forme dans une certaine mesure, permettant ainsi de dépasser le cadre strict des formes prévues initialement.

Méthodes d'apprentissages

Les méthodes d'apprentissages consistent à extraire lors d'une étape de pré-process des informations spécifiques des formes données en entrée, et à appliquer des déformations prenant en compte ces informations dans une seconde étape.

Cette approche est issue de la généralisation de l'interpolation de forme. Elle possède l'avantage de ne pas devoir stocker en permanence l'ensemble des formes de bases, et n'en retient que les informations principales si certaines sont redondantes d'un modèle à l'autre.

Afin de s'appliquer à l'animation de personnages articulés, certaines approches se basent spécifiquement sur des a-priori géométriques issus du skinning, puis améliorent ce modèle à partir des informations issues de l'apprentissage. Certaines approches extraient des informations purement numériques issues de décomposition en composantes principales [LCF00, KJP02, WP02] permettant de gérer des déformations arbitraires, mais reste très peu intuitif à manipuler. D'autres approches comme celles de Yang *et al.* [YPP07] fittent des fonctions de déformations associées à la pose géométriques par régression, ce qui rend le modèle moins générique, mais plus aisément manipulables. Un exemple illustré est montré en fig. 1.38.

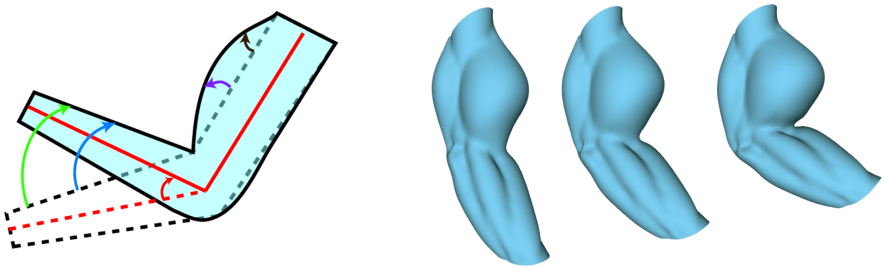


FIGURE 1.38 – *Illustration d'une méthode d'apprentissage appliquée sur un cas de bras skinné. Une régression préliminaire lie l'angle de rotation d'un joint à l'orientation des triangles composant le maillage. Les auteurs [YPP07] parviennent ainsi à mimer l'effet de gonflement musculaire lors de l'articulation du bras.*

Récemment, les approches d'apprentissages permettent également de modéliser en temps réel des animations de vêtements détaillés présentant des plis [KSO10, WHRO10, dASTH10].

Avantages et inconvénients

Comme nous avons pu voir, les déformations basées exemples permettent l'animation rapide et plausible de surfaces. L'utilisation de telles méthodes est cependant liée à deux problématiques principales

1. Le contrôle artistique de la forme obtenue.
2. La création des formes de clés.

En effet, les formes générées par de telles méthodes doivent appartenir à l'espace possible défini par les formes de bases ou d'apprentissage. Par exemple, l'apprentissage basé sur un ensemble complet de personnages qui marchent sera incapable de générer un mouvement de course. Il est ainsi nécessaire de définir à l'avance l'ensemble des déformations possibles ce qui rend ces approches peu adaptées à des déformations quelconques. Ce type d'approche est donc inadapté à la manipulation d'une forme interactivement comme présenté en partie 2.7.

Pour obtenir des degrés de liberté supplémentaires, il est alors nécessaire de modéliser un grand nombre de pauses d'apprentissages. Il existe deux grandes sources de génération de telles pauses.

- La première méthode consiste à capturer le plus d'information possible à partir de données réelles : reconstructions par caméra, motion capture, etc. Il faut cependant déployer une infrastructure lourde et couteuse pour mettre en place ce type de capture. De plus, une fois capturée, la modification de ces modèles n'est plus réalisable si un défaut est détecté. Il est alors nécessaire de relancer une capture. Enfin, les modèles associés à cette approche sont nécessairement des personnages aux formes proches de celles du modèle capturé.
- La seconde approche, souvent couplée à la première, consiste à modéliser (ou ajuster) manuellement l'ensemble des poses de bases. C'est un travail excessivement fastidieux et gourmand en temps qui nécessite l'intervention de plusieurs artistes. C'est également la seule méthode capable de générer des monstres dont l'aspect s'éloigne trop de dimensions anthropomorphiques [WP02].

La minimisation du nombre de poses d'apprentissage nécessite la mise en place d'extraction de données avancées basées par exemple sur les décompositions en éléments principaux. Cependant, les *formes de bases* à combiner issues de ces modèles numériques ne correspondent généralement plus à des formes physiques habituelles. Leur manipulation n'est alors plus intuitive.
De plus, la mise en place d'un système d'apprentissage sur des pauses non adaptées risque de générer des déformations non souhaitées.
Par exemple, si l'on suppose que dans l'ensemble des pauses d'apprentissages les personnages bougent le bras gauche et droit de manière symétrique, le modèle numérique intégrera une corrélation non réaliste entre le mouvement de ces deux bras. Au final, l'animation du personnage sera contrainte de respecter ce principe de symétrique présent dans les modèles d'apprentissages.

Bien que le développement des méthodes de captures de scènes réelles permette d'intégrer des animations de réalisme croissant, nous n'utiliserons pas dans ce travail d'approche de ce type. Ainsi, nous ne serons pas dépendant de données d'entrées longues et complexes à en place, et nous pourrons garder un contrôle géométrique intuitif vis-à-vis des déformations apportées.

1.3.3 Déformation basée surfaces

Un dernier type de méthode de déformation consiste à analyser la surface elle-même afin d'en déduire la transformation à réaliser. L'intérêt principal de l'approche est de pouvoir adapter le type de déformation en fonction des caractéristiques propres de la surface. Une surface plane ne sera ainsi pas déformée de la même manière qu'une surface présentant des arêtes saillantes.

Ces méthodes ont l'avantage de permettre le contrôle direct de la surface. Bien que moins adaptées à l'animation de personnages articulés que le skinning ou à l'animation de vêtement que la simulation, elles sont particulièrement efficaces dans un contexte de modélisation de formes libres.

Nous présentons deux applications typiques, encore une fois, sans chercher d'exhaustivité :

1. La déformation multirésolution par subdivision.
2. L'utilisation de coordonnées différentielles.

Le premier cas est une méthode devenue standard en graphique en permettant la modélisation multi-échelle ; alors que la seconde application, plus récente, permet la prise en compte des propriétés de courbure différentielles de la surface.

Déformation par surfaces de subdivision

L'intérêt principal des surfaces de subdivision dans le cadre de la déformation et l'animation consiste en la capacité à manipuler la forme grossière d'un objet représenté par un maillage, et de propager cette information de déformation à l'ensemble des résolutions plus fines.

Il est ainsi possible de déformer la surface à différents niveaux de détails, de l'orientation d'une partie du corps d'un personnage, à la déformation locale liée à un muscle.

Principe

La modélisation par surfaces de subdivisions permettent de résoudre le problème de gestion de la topologie des surfaces polynomiales définies par produit tensoriel (partie 1.1.1) de manière simple et rapide.

Les surfaces de subdivisions fonctionnent sur un principe itératif en deux étapes :

1. Subdivision de la connectivité du maillage, indépendamment du plongement géométrique (fig. 1.39).
2. Choix de la position pour les anciens et les nouveaux sommets basés sur l'information locale des coordonnées des anciens sommets (fig. 1.40).

En itérant cette subdivision à partir d'un calcul de position judicieusement choisie, on peut montrer que l'on peut obtenir la convergence du maillage vers une surface ayant des propriétés de continuité C^1 voir C^2.

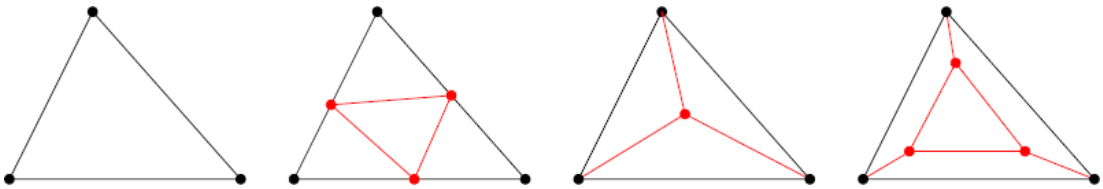


FIGURE 1.39 – *Différentes possibilitées de subdivision de la connectivité d'un triangle.*

Méthodes pré-existentes

Défini initialement pour la modélisation de surfaces lisses dans une optique de CAO [CC78, Loo87], les surfaces de subdivisions furent introduites pour la première fois dans une application d'animation par De Rose *et al.* [DKT98] pour la production du court-métrage Géri de Pixar [9]. Pour cela, les auteurs étendent l'étape de positionnement des nouveaux sommets pour

9. http ://www.pixar.com/

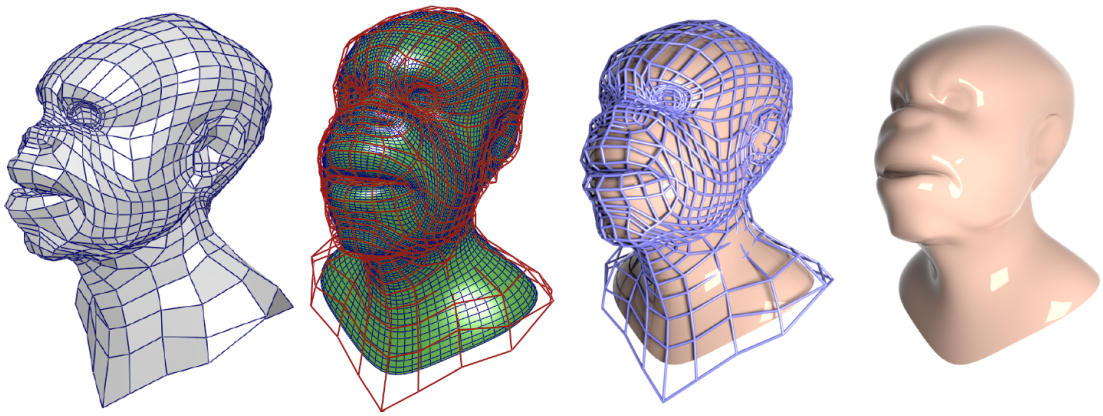


FIGURE 1.40 – *De gauche à droite : polygone de contrôle ; polygone de contrôle et maillage subdivisé ; rendu du polygone de contrôle et de la surface ; surface subdivisée finale.*

prendre en compte la gestion des collisions, ainsi que la possibilité de garder certaines arêtes vives.

Les surfaces de subdivisions sont la base de nombreux outils de modélisation, et une littérature abondante est disponible. Nous invitons le lecteur intéressé à consulter l'état de l'art proposé par Zorin et Schröder [ZSD+00]. Nous noterons également sont utilisation dans l'application de déformations rapides [ZSS97].

Inconvénients

La modélisation et l'animation par subdivision présentent certains inconvénients

- Premièrement, les surfaces de subdivisions n'ont pas de paramétrisation sous-jacente connue, ou du moins celle-ci est complexe à calculer. Il s'agit de la raison principale pour laquelle l'utilisation de ces surfaces n'est toujours pas généralisée dans les logiciels de CAO. En effet, les opérations classiques telles que la découpe, les intersections et les raccordements nécessitent cette connaissance.
- Deuxièmement, la forme finale obtenue après N subdivisions est difficilement prédictible. En effet, celle-ci dépend non seulement des positions des sommets, mais également de la connectivité. Ainsi, deux maillages modélisant une surface identique, mais dont les connectivités sont différentes ne convergeront pas vers la même surface lisse (voir fig. 1.41). L'artiste modélisant à l'aide de cette approche perd donc une part de contrôle et doit à la fois jouer par essais-erreurs tout en possédant la connaissance technique lui permettant d'interpréter l'origine de ce phénomène.
- En corrolaire de la précédente remarque, le maillage basse résolution ne fournit que peu d'information par rapport à la surface finale. En particulier, l'aire et le volume de la surface finale ne sont pas exprimables analytiquement.
- Enfin, le principe de subdivision est particulièrement adapté pour s'ajouter en temps que post-process permettant d'améliorer la qualité visuelle du résultat. Elle s'avère moins pratique pour définir des outils de déformations de par les contraintes non linéaires et difficilement paramétrables d'un niveau de subdivision à l'autre. Il n'est donc pas trivial de déformer le maillage à basse résolution pour donner l'effet escompté sur la surface finale après convergence.

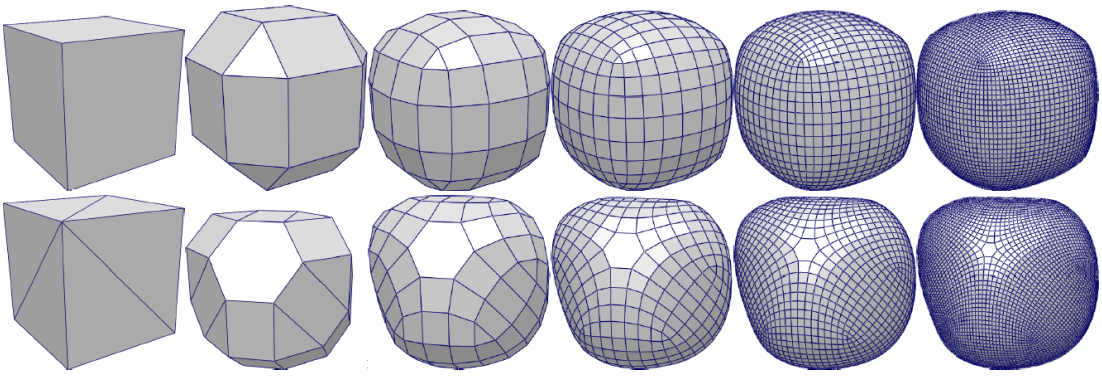


FIGURE 1.41 – *Dépendance du processus de subdivision à la connectivité du maillage. Un même processus de subdivision est ici appliqué sur deux polygone de contrôle partageant la même géométrie mais une connectivité différente. Les processus de subdivisions montrés de gauche à droite convergent vers des surfaces d'aspects différents.*

Nos méthodes de déformation ne se baseront pas sur le principe fondamental de la subdivision. Cependant, ce type d'algorithme pourrait venir se rajouter en tant que post-process supplémentaire afin d'améliorer le résultat visuel final.

Déformations par coordonnées différentielles

Récemment, une méthode d'encodage de surface maillée s'est avérée particulièrement adapté à l'application de déformations quelconques. La mise en place de système d'encodage différentiel tente de prendre avantage des notions de géométries différentielles étendues au cas de maillages polygonaux. Sans entrer dans les détails de ce domaine, nous pourrons référer le lecteur intéressé vers l'état de l'art de Grinspun et Wardetzky [GWDS08].

Principe

Avec cette approche, la surface n'est plus définie par ses coordonnées cartésiennes standard, mais par une représentation intrinsèque ne dépendant pas de sa position dans l'espace.

Un exemple de tel codage dénommé *coordonnées Laplaciennes* décrit la surface localement de manière différentielle.

Comme proposé dans [Sor06], la représentation la plus simple des coordonnées *Laplaciennes* consiste à considérer qu'un sommet i de 1-voisinage $\mathcal{V}_i$ de cadinal $N_{\mathcal{V}_i}$ est associé à la coordonnée $\boldsymbol{\delta}_i$ avec

$$\boldsymbol{\delta}_i = \mathbf{p}_i - \frac{1}{N_{\mathcal{V}_i}} \sum_{k=1}^{N_{\mathcal{V}_i}} \mathbf{p}_k \,. \tag{1.19}$$

En d'autres termes, $\boldsymbol{\delta}_i$ est le vecteur mesurant la différence entre le sommet courant, et le barycentre du 1-voisinage. De manière intuitive, cette mesure est liée à la courbure moyenne de la surface en ce point [MDSB02]. Cette approximation de l'opérateur de Laplace-Beltrami sur une variété fut introduite initialement par Taubin [Tau95].

On notera également que la relation entre les coordonnées différentielles et absolues peut s'exprimer linéairement

$$\begin{pmatrix} \boldsymbol{\delta}_0 \\ \vdots \\ \boldsymbol{\delta}_{N_s} \end{pmatrix} = \mathrm{M} \begin{pmatrix} \mathbf{p}_0 \\ \vdots \\ \mathbf{p}_{N_s} \end{pmatrix} .$$

L'idée de la déformation est alors la suivante. Soit $\mathbf{p}_i^0$ la position absolue du sommet i du maillage non déformé. δ_i^0 les coordonnées différentielles de ce même sommet. On vient alors déplacer un ensemble quelconque de sommets du maillage original pour lui donner une position et orientation proche de celle souhaitée. Soit $\mathcal{I}$ l'ensemble des indices de ces sommets manipulés, et $\mathbf{p}_i^c$ les coordonnées de ces sommets.

On cherche alors les nouvelles coordonnées différentielles $\boldsymbol{\delta}_i^1$ et nouvelles positions absolues $\mathbf{p}_i^1$ tel que

$$\sum_{\substack{i=1 \\ i \notin \mathcal{I}}}^{N_s} \|\boldsymbol{\delta}_i^1 - \boldsymbol{\delta}_i^0\|^2 + \sum_{j \in \mathcal{I}} \|\mathbf{p}_j^1 - \mathbf{p}_j^c\|^2$$

soit minimal.

En utilisant la propriété de linéarité des coordonnées différentielles, on peut alors écrire le système sous la forme d'un système global matriciel creux à minimiser. Des algorithmes efficaces de résolution de systèmes linéaires creux permettent alors d'obtenir le maillage déformé. Les sommets de celui-ci approximeront à la fois les contraintes de position souhaitées par l'utilisateur tout en préservant le mieux possible [10] les propriétés différentielles apportées par les coordonnées Laplaciennes comme illustrées en fig. 1.42.

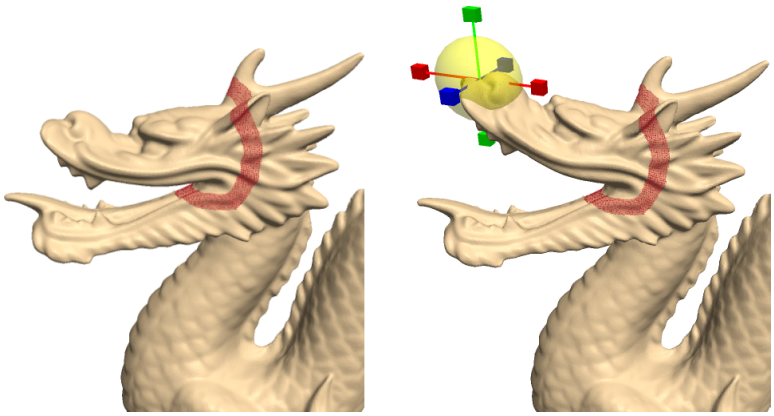


FIGURE 1.42 – *Exemple de déformation interactive proposée dans [SCOL+04] basée sur les coordonnées différentielles.*

Méthodes existantes

Introduites par Lipman *et al.* [LSCO+04] les coordonnées différentielles de l'eq. (1.19) ne sont malheureusement pas invariantes par rotation. Sorkine *et al.* [Sor06] et Masuda *et*

10. au sens des moindres carrés

al. [MYF06] suggèrent plus tard l'utilisation de coordonnées dites des *cotangentes* pour approximer le *Laplacien* du maillage plus précisément en se basant sur une méthode introduite initialement par Pinkall et Polthier [PP93]. Wardesky *et al.*[WMKG07] montrent cependant qu'il ne peut exister d'expression discrète d'un Laplacien possédant l'ensemble des propriétés souhaitées dans le cas continu [11].

Il en résulte que l'utilisation directe de cette expression ne permet pas de gérer correctement des maillages dont certaines parties ont subi, ou doivent subir l'application de rotations. Yu *et al.* [YZX+04, XZWB05, XZB06] introduisent une approche quasi-similaire aux coordonnées Laplaciennes en se basant sur une équation de diffusion de Poisson, mais n'est pas invariant par rotation non plus.

Pour tenter de prendre en compte la partie rotationnelle de la transformation, Sorkine *et al.* [SCOL+04] introduisent une étape de minimisation supplémentaire sur l'application de rotations. Lors de l'utilisation d'angles de rotations importants, il est cependant nécessaire d'itérer à plusieurs reprises la résolution d'un système linéaire creux global ce qui rend la méthode plus couteuse à mettre en place.

Sheffer et Kraevoy [SK04] présentent un encodage différentiel intrinsèquement invariant par rotation, mais nécessite une reconstruction non linéaire du maillage, alors que Lipman *et al.* [LSLCO05] proposent une autre définition préservant l'invariance par rotation et la linéarité. Cependant, contrairement à l'utilisation de coordonnées Laplaciennes, ces méthodes requièrent explicitement que l'utilisateur indique la rotation à appliquer. En effet, la présence de translations locales empêchent alors de trouver automatiquement les composantes des rotations.

1.4 DÉFORMATION SOUS CONTRAINTES

Nous avons décrit un certain nombre de déformations standard de surfaces pouvant servir à l'animation de personnages et objets virtuels. Comme nous l'avons vu, il est généralement nécessaire de réaliser des compromis entre

- Liberté artistique,
- Rapidité d'évaluation,
- Garantie d'une plausibilité visuelle.

Les méthodes géométriques de skinning ou de simulations physique à basse résolution permettent de répondre aux deux premiers points, mais ne garantissent pas la dernière propriété. En effet, certaines propriétés géométriques de la surface telles que le maintien du volume englobé, de son aire ou de ses longueurs vont se trouver perdues au cours de l'animation.
Afin de limiter la tâche fastidieuse de correction manuelle pour tenter de minimiser ces derniers points, plusieurs méthodes ont déjà été développées. Nous détaillons certaines d'entres-elles.

1.4.1 Contraintes de volumes

De nombreux objets réels gardent un volume constant pendant qu'ils sont déformés. Typiquement les membres des personnages et animaux gardent localement un volume quasi constant de par leur structure interne [12] (le corps étant à 80% constitué d'eau). Dans le cas d'un objet virtuel représenté par sa surface, la préservation du volume englobé par celle-ci est reconnue pour jouer un impact considérable vis-à-vis de son réalisme [Las87].

11. Linéarité, positivité, symétrie, localité, ...
12. À l'exception du thorax pendant la respiration

L'information de volume englobée n'est cependant pas directement accessible. De plus, les méthodes de déformation standard telles que le *skinning* ne préservent pas celui-ci. Certaines méthodes pré-existentes proposent ainsi d'ajouter une contrainte géométrique pour préserver le volume d'un objet défini par sa surface.

Transformations préservant le volume

Étant donné un compact Ω de $\mathbb{R}^3$ de volume V, on peut distinguer trois grandes approches de méthodes préservant le volume.

1. La préservation numérique de la valeur du volume V.
2. L'application d'une déformation dont le Jacobien vaut 1.
3. Une déformation par champ de vecteur à divergence nulle.

Nous présenterons tout d'abord le principe de ces approches, puis nous détaillerons dans la suite des exemples d'utilisation en partie 1.4.2.

Préservation numérique du volume Considérons une surface S représentant le bord d'un objet volumique. Mathématiquement, S est une surface de $\mathbb{R}^3$ compacte, fermée, plongée sans auto-intersection. C'est-à-dire que S représente le bord d'un sous volume connexe de $\mathbb{R}^3$. On notera $\partial\Omega = S$ le bord du domaine Ω.

Le volume V du domaine Ω est défini par

$$V = \int_{\Omega} \mathrm{d}\Omega \,. \tag{1.20}$$

Notons que garder V constant ne conserve le volume que globalement ce qui n'est pas forcément suffisant selon les applications. Nous utiliserons cette approche géométrique au chapitre 2 mais l'étentrons à la préservation de volume locale.

Jacobien d'une déformation Supposons une déformation bijective C^1 de l'espace

$$g : \left\{ \begin{array}{ccc} \mathbb{R}^3 & \rightarrow & \mathbb{R}^3 \\ \mathbf{p} & \mapsto & g(\mathbf{p}) \end{array} \right.$$

g préserve localement le volume de toute partie de l'espace si son Jabobien $\det(\mathrm{J})$ vaut 1, avec J la matrice des dérivées partielles telle que

$$\mathrm{J} = \left[\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}, \frac{\partial g}{\partial z} \right] \,.$$

La préservation locale de volume implique sa préservation globale sur l'ensemble d'un domaine. En effet, le changement de variable de $\mathbf{p}$ en $g(\mathbf{p})$ dans l'eq. (1.20) fait uniquement intervenir son jacobien. Celui-ci valant 1, la valeur de l'intégrale n'est pas modifiée.

Champ de vecteurs à divergence nulle Considérons une déformation définie par l'intégration le long d'un champ de vecteur telle que présenté en partie 1.3.1. Si le champ $\mathbf{v}$ est à divergence nulle (également appelé champ incompressible, ou champ solénoidale), c'est à dire tel que

$$\forall \mathbf{x} \in \mathbb{R}^3, (\nabla \cdot \mathbf{v})(\mathbf{p}) = 0 \,,$$

alors les lignes de champs ne se croisent jamais. Cela implique que la surface déformée ne s'auto-collisionne pas. De plus, si la surface considérée définit le bord d'un domaine fermé de l'espace, alors celui-ci sera déformé sans modification de son volume.

L'utilisation d'un tel champ de déformation implique ainsi que le volume soit préservé localement. Il s'agit d'une propriété largement utilisée pour la simulation de fluide incompressible.

Ces trois relations ont déjà été utilisées pour définir des méthodes de déformations à volume constant que nous détaillerons en section 1.4.2. On notera que d'une manière générale, les contraintes de préservation de volume locale sont très restrictives (par exemple, le Jacobien impose de travailler par déformation de l'espace), il peut ainsi être difficile de définir des méthodes de déformations générales laissant suffisamment de liberté artistique. Au contraire, la contrainte de préservation numérique de volume globale peut s'adapter à tout type de déformation tel que le skinning, tout en laissant une grande liberté pour affiner le résultat visuel suivant l'effet souhaité. Les difficultés principales étant de mettre en place des méthodes de contrôle sur la forme obtenue, et de localiser la déformation aux endroits souhaités.

Le paragraphe suivant explique comment calculer le volume englobé par une surface fermée quelconque dans le cadre de l'utilisation d'un volume de contrôle global.

Expression du volume

Nous présentons tout d'abord le calcul du volume associé à un objet bordé par une surface lisse, puis dans le cas particulier d'un maillage polygonal.

En considérant une surface S suffisamment régulière paramétrée par (u, v) sur le domaine $\mathcal{D}$, on peut écrire [GOCP97] (la démonstration est fournie en annexe A.1) l'expression du volume comme intégrale de surface :

$$V = \iint_{(u,v)\in\mathcal{D}} < \mathbf{p}(u,v), \mathbf{e} >< \mathbf{n}(u,v), \mathbf{e} > \, \mathrm{d}u \, \mathrm{d}v \tag{1.21}$$

où $\mathbf{n}$ correspond à la normale orientée de la surface (sortante par rapport à Ω) au point $\mathbf{p}$, et $\mathbf{e}$ est un vecteur unitaire quelconque.

Pour des raisons de simplicité, un cas particulier fréquent consiste à considérer $\mathbf{u}$ comme étant aligné avec l'axe des coordonnées z, c'est à dire $\mathbf{e} = \mathbf{u}_z$. On a alors

$$V = \iint_{(u,v)\in\mathcal{D}} z(u,v)\, n_z(u,v) \; \mathrm{d}u\mathrm{d}v \,, \tag{1.22}$$

avec $n_z :=< \mathbf{n}, \mathbf{u}_z >$.

On peut enfin montrer (voir annexe A.2) que dans le cas où S est un maillage triangulaire fermé, le volume s'exprime sous la forme suivante [HML99]

$$\begin{cases} V = \dfrac{1}{6}\displaystyle\sum_{i=1}^{N_p}(z_{a_i} + z_{b_i} + z_{c_i}) \begin{vmatrix} x_{b_i} - x_{a_i} & x_{c_i} - x_{a_i} \\ y_{b_i} - y_{a_i} & y_{c_i} - y_{a_i} \end{vmatrix} \\ \\ \forall i \in [\![1, N_p]\!] \ , \quad \mathcal{F}_i = (a_i, b_i, c_i) \ , \end{cases} \tag{1.23}$$

où $\mathcal{F}_i$ désigne la face i.

Interprétation géométrique Cette expression possède une interprétation géométrique. Premièrement, l'eq. (1.23) peut s'écrire sous la forme

$$V = \sum_{\text{triangles}} z_{\text{avg}}\, \mathcal{A} \ , \tag{1.24}$$

où $z_{\text{avg}} = 1/3(z_{a_i} + z_{b_i} + z_{c_i})$ est la hauteur moyenne du triangle considéré, et $\mathcal{A}$ est l'aire du triangle courant projetée sur le plan d'équation $z = 0$:

$$\mathcal{A} = \frac{1}{2}(\mathbf{p}_{b_i} - \mathbf{p}_{a_i}) \times (\mathbf{p}_{c_i} - \mathbf{p}_{a_i}) \cdot \mathbf{u}_z \ .$$

Chaque sous volume de la somme correspond au volume d'un prisme obtenu en extrudant le triangle courant sur le plan d'équation $z = 0$ comme illustré en fig. 1.43. En sommant algébriquement l'ensemble de ces sous-volumes signés, on obtient ainsi le volume total délimité par le maillage.

Rem. Il est également possible de privilégier l'axe x ou y dans les formules (1.22) et (1.23). Notons que cette volume n'est valide que dans le cas où le maillage est fermé.

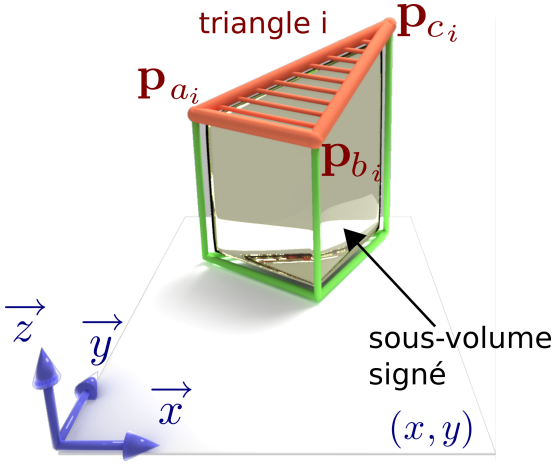


FIGURE 1.43 – *Illustration géométrique l'expression du volume donné en eq. (1.23). Le triangle courant $(\mathbf{p}_{a_i}, \mathbf{p}_{b_i}, \mathbf{p}_{c_i})$ est extrudé le long de l'axe z jusqu'à plan d'équation $z = 0$. Le volume signé du prisme correspond à la contribution associée à un triangle.*

Trilinéarité On notera également que le volume s'exprime de manière trilinéaire en fonction des coordonnées des sommets du maillage. Nous utiliserons cette propriété particulière dans la partie 2.3.2.

Notons $X \in \mathbb{R}^{N_s}$ (respectivement Y et Z) le vecteur concaténé des coordonnées x (resp. y et z) de tous les sommets du maillage. On peut alors écrire le volume sous forme d'un produit scalaire dans $\mathbb{R}^{N_s}$

$$V =< \nabla_{\mathbf{x}} V, X >=< \nabla_{\mathbf{y}} V, Y >=< \nabla_{\mathbf{z}} V, Z > \quad , \tag{1.25}$$

où $\nabla_{\mathbf{x}} V \in \mathbb{R}^{N_s}$ est le vecteur des dérivées partielles de V par rapports aux coordonnées x des sommets, et similairement pour $\nabla_{\mathbf{y}} V$ et $\nabla_{\mathbf{z}} V$.

1.4.2 Méthodes de déformation à volume constant

Nous présentons ici différentes méthodes développées afin de préserver le volume défini par une surface.

Deformation de l'espace

Free Form Deformation

Comme présentées en partie 1.3.1, les méthodes de FFD furent parmi les premières méthodes de déformation géométrique. La mise en place de contraintes de déformations à volume constant a été étudiée de différentes façons pour ce type d'approche.

[RSB96] définissent une déformation de forme libre avec préservation de volume. Leur expression de volume est directement adaptée au cas d'une déformation par fonction de Bézier. Soit S_1 la surface obtenue après déformation par FFD. S_0 étant la surface originale, et S_2 étant la surface cherchée telle que $V(S_2) = V(S_0)$. Les auteurs définissent la surface S_2 par minimisation sous contrainte s'exprimant sous la forme

$$\begin{cases} \min \sum_{k=1}^{N_c} \|\mathbf{c}_1^k - \mathbf{c}_2^k\| \\ \text{tel que } V(S_2) = V(S_0) \, , \end{cases}$$

avec $\mathbf{c}_1$ (respectivement $\mathbf{c}_2$) représentant les points de contrôle de la grille FFD associé à la surface S_1 (respectivement S_2), et N_c étant le nombre de points de controle. La méthode des multiplicateurs de Lagrange [QSS00] est utilisée pour résoudre numériquement le système sous contrainte non linéaire. La méthode utilisée reste cependant excessivement lourde.

Hirota *et al.* [HML99] proposent également une déformation de FFD préservant globalement le volume (voir fig. 1.44). Ils se basent cette fois sur l'expression analytique du volume donné par l'eq. (1.23). Soit $P_0 \in \mathbb{R}^{3\,N_s}$ le vecteur des coordonnées (x, y, z) des sommets concaténés avant déformation par FFD. Soit P_1 le vecteur après déformation par FFD. Les auteurs cherchent alors un ensemble de coordonnées P_2 tel que

$$\begin{cases} \min E_{\text{spring}}(S_1, S_2) \\ \text{tel que} V(P_2) = V(P_0) \, , \end{cases}$$

avec $E_{\text{spring}}(S_1, S_2)$ correspondant à l'énergie d'un ensemble de ressorts (voir partie 1.2.3) reliant les sommets de la grille de contrôle et dont les longueurs aux repos correspondent à ceux obtenus après déformation par FFD classique. L'expression reste non linéaire, et l'utilisation d'un optimiseur global est toujours nécessaire. Afin d'accélérer la convergence, les auteurs

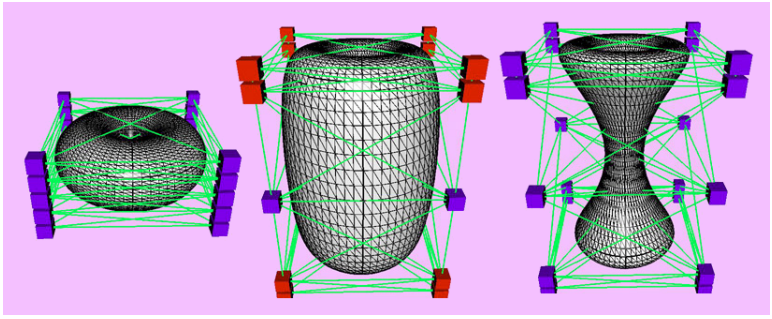


FIGURE 1.44 – *Déformation par FFD à volume constant. Le modèle initial est montré à gauche. Certains points de contrôles sont modifiés (milieu) et les points de contrôle fixes sont colorés en rouge. Après application de la préservation de volume, la nouvelle grille et la figure résultante sont montrées à droite. Image tirée de [HML99].*

utilisent une approche multiresolution. En initialisant l'optimisation à basse résolution, puis en raffinant au fur et à mesure le maillage, les auteurs parviennent à limiter le coût de calcul.

On notera cependant que pour ces deux approches, le volume est uniquement globalement préservé et non localement. Ainsi, une déformation localisée peut avoir une répercussion sur la forme globale et n'est donc pas adaptable à l'animation de personnages articulés. De plus, l'utilisateur n'a pas de contrôle sur la forme finale issue de l'optimiseur qui peut être arbitrairement éloignée de celle souhaitée.

Notons que les extensions de la FFD sur des cages arbitraires peuvent également se voir doter de contrainte de préservation de volume. Cependant, de par la nature non linéaire de ces coordonnées, seules des minimisations non linéaires globales sont étudiées comme dans l'approche de Ben-Chen *et al.* [BCWG09].

Manipulation par contraintes

On rappelle que dans le cas où la surface est déformée suivant une manipulation directe (voir partie 1.3.3) d'un sous-ensemble des sommets, le reste du maillage est généralement reconstruit à l'aide d'une fonction à minimiser globalement. Celle-ci peut s'exprimer de manière linéaire ou non.

Plusieurs approches ont déjà permis d'incorporer des contraintes de préservation de volume lors de ces déformations afin d'améliorer le résultat visuel.

Premièrement, Huang *et al.* [HSL+06] et Shi *et al.* [SZT+07] introduisent des contraintes de préservation globale de volume et de rigidités issues de la présence d'un squelette d'animation interne dans la résolution du système de coordonnées différentielles. Ils limitent ainsi une perte visuelle volumique trop importante, mais ne préservent cette contrainte que de manière approchée à l'aide d'une minimisation globale. Zhou *et al.* [ZHS+05] proposent quant à eux de construire un modèle tétraédrique à partir de l'information de la surface, et d'appliquer la minimisation Laplacienne directement sur le volume du maillage. L'approche permet cette fois de limiter les déformations volumiques locales, mais ne préserve pas le volume et reste incapable de reproduire un effet de gonflement musculaire (voir fig. 1.45).

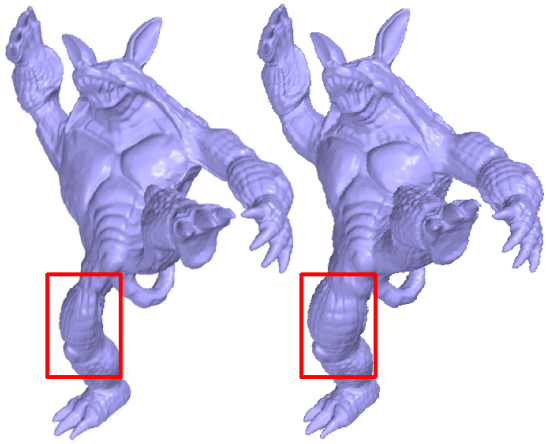


FIGURE 1.45 – *Déformation à volume constant par déformation Laplacienne volumique. Gauche : sans preservation de volume, droite avec. Image de [ZHS+05].*

Enfin, une approche originale est proposée par Lipman *et al.* [LCORL07]. L'approche spécifiquement conçue pour gérer les grandes rotations consiste à paramétrer un champ de rotation volumique. Le volume local étant alors exprimé en fonction des variations de la normale à la surface. L'approche nécessite toujours la résolution d'un large système linéaire creux à chaque étape.

Déformation quelconque pré-existante

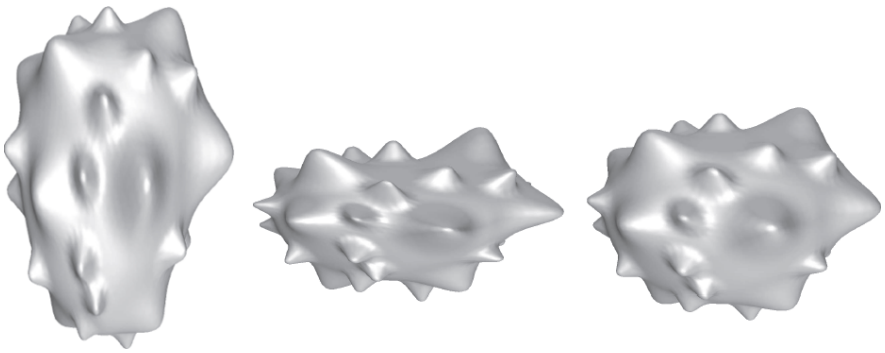


FIGURE 1.46 – *Déformation multiresolution à volume constant. Gauche : surface originale. Milieu : déformation sans préservation de volume.Droite : déformation avec préservation de volume. Image de [SHB07].*

Dans une optique plus générale, Sauvage *et al.* [SHB07] définissent une méthode rapide de préservation de volume appliqué en tant que post-correction sur un maillage déformé de

manière quelconque (voir fig. 1.46). En particulier, ils montrent qu'en linéarisant l'expression de la contrainte de volume, on peut définir une solution analytique approchée sans avoir recours à une minimisation numérique.

Nous détaillons ci-après la méthode de résolution, car nous nous inspirerons de celle-ci dans notre application de skinning à volume constant.

Minimisation On considère de manière générale P_0 et P_1 comme étant les positions concaténées des sommets du maillage original et après déformation. L'approche est la suivante : On cherche un vecteur de déformation par sommet $U = (\mathbf{u}_1, \ldots, \mathbf{u}_{N_s})$ tel que $\mathbf{P}_2 = \mathbf{P}_1 + \mathbf{U}$ avec

$$\begin{cases} \min & \|\mathbf{U}\|^2 \\ \text{contraint à} & V(\mathbf{P}_1 + \mathbf{U}) = \mathbf{V}(P_0) \ . \end{cases} \tag{1.26}$$

Afin de résoudre ce problème sous contraintes, les auteurs utilisent la méthode des multiplicateurs de Lagrange. On introduit pour cela un paramètre scalaire supplémentaire λ et l'on réécrit ce problème sous la forme d'une fonctionnelle d'énergie :

$$E(\mathbf{U}, \lambda) = \|\mathbf{U}\|^2 + \lambda \left(V(\mathbf{P}_1 + \mathbf{U}) - V(\mathbf{P}_0)\right) \ .$$

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

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

Si la solution de ce problème existe, alors il est nécessaire que celle-ci annule le gradient de E (extremum de E) : $\nabla E(\mathbf{U}, \lambda) = 0$.

Solution linéarisée La contrainte de volume est une fonction trilinéaire suivant les coordonnées (x, y, z) de $\mathbf{U}$ que nous noterons $\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z$. Pour pouvoir résoudre analytiquement ce problème de minimisation, il est possible de transformer cette contrainte cubique en une expression approchée linéaire.

On suppose que S_2 est très proche de S_1. Ainsi, on considère $\|\mathbf{U}\|$ suffisamment petit pour linéariser l'expression du volume. On peut alors écrire en utilisant les notations de l'eq. (1.25) :

$$\begin{aligned} V(\mathbf{P}_1 + \mathbf{U}) - V(\mathbf{P}_0) \quad &\simeq V(\mathbf{P}_1) - V(\mathbf{P}_0) + \\ &< \nabla_{\mathbf{x}} V, \mathbf{U}_x > + < \nabla_{\mathbf{y}} V, \mathbf{U}_y > + < \nabla_{\mathbf{z}} V, \mathbf{U}_z > \ . \end{aligned} \tag{1.27}$$

En notant $\Delta V = V(\mathbf{P}_1) - V(\mathbf{P}_0)$ et en considérant que l'on corrige le volume de manière uniforme suivant l'axe x, y et z, la solution analytique du vecteur de déformation à appliquer s'exprime sous la forme (voir annexe B.1) :

$$\forall i \in [\![1, N_s]\!] \ , \quad \mathbf{u}_i = \frac{\Delta V}{3} \left(\frac{\nabla_{\mathbf{x}} V(\mathbf{p}_i)}{\|\nabla_{\mathbf{x}} V\|^2}, \frac{\nabla_{\mathbf{y}} V(\mathbf{p}_i)}{\|\nabla_{\mathbf{y}} V\|^2}, \frac{\nabla_{\mathbf{z}} V(\mathbf{p}_i)}{\|\nabla_{\mathbf{z}} V\|^2} \right) \ , \tag{1.28}$$

avec $\nabla_{\mathbf{x}} V(\mathbf{p}_i)$ représentant la $i$ème coordonnée du vecteur $\nabla_{\mathbf{x}} V$,

$$\|\nabla_{\mathbf{x}} V\| = \sqrt{\sum_{i=1}^{N_s} \left(\nabla_{\mathbf{x}} V(\mathbf{p}_i)\right)^2} \ ,$$

et similairement pour x et y.

Discussion Cette approche permet d'appliquer une déformation secondaire connue analytiquement sur chaque sommet d'un maillage permettant de garder un volume approximativement constant lors d'une déformation. L'approche est suffisamment générale pour s'appliquer à tout type de déformation.

Sauvage [Sau05] propose également une méthode de correction itérative exacte se basant sur la même idée, mais ne nécessitant pas d'approximation. Cette méthode est appliquée sur une préservation d'aire de surface de Bézier, mais se généralise au cas de la préservation de volume d'un maillage.

Cependant, comme l'ensemble de ces approches, celle-ci reste globale et n'offre pas de contrôle utilisateur afin de maitriser l'apparence finale après correction de volume. Dans le chapitre 2, nous reprendrons cette idée de préservation en introduisant la localisation de l'évaluation et de la préservation de volume, ainsi qu'un contrôle utilisateur de la forme finale.

Interpolation de rotations

Nous avons présenté en partie 1.3.1 la méthode de déformation par le geste basée sur l'interpolation maticielle appelée *Sweepers* proposée par Angelidis *et al.* [AWC04]. Les auteurs proposent [ACWK04a, ACWK04b] une extension de cette méthode à la modélisation à volume constant.

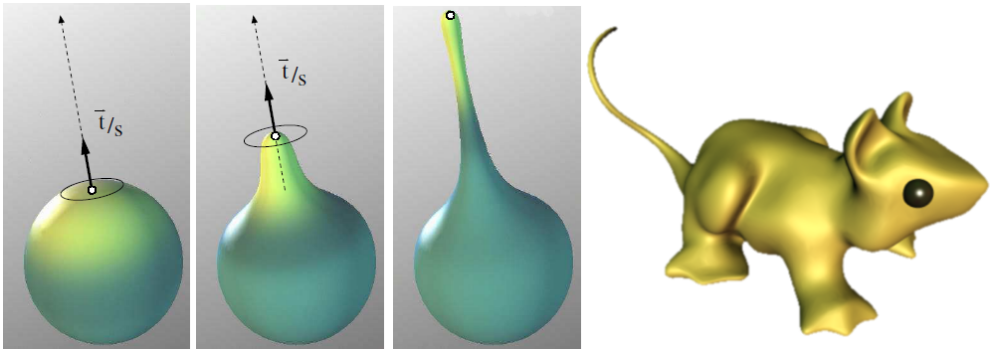


FIGURE 1.47 – *Déformation à volume constant tirée de [ACWK04b]. Les images de gauches représentent l'outil de déformation préservant le volume. L'image de droite représente une réalisation formée à partir d'une sphère unique.*

Pour cela, ils interpolent un type particulier de déformation appelé *swirl*. Un *swirl* correspondant à un champ de rotation dont l'angle de rotation décroit en fonction de l'éloignement au centre de l'application de cette déformation. En combinant adéquatement un ensemble de rotations, les auteurs définissent un outil capable de déformation permettant de mimer l'apparence d'une modélisation de type pâte à modeler. En particulier, les auteurs montrent qu'une déformation par *swirl* est caractérisée par une déformation dont le Jacobien est égal à 1, et prouvent ainsi que la méthode préserve le volume localement.

Ce type d'outil se révèle particulièrement bien adapté pour de la modélisation de forme par sculpture. Par contre, elle ne s'adapte pas directement au cas de l'animation de personnage articulé dont l'apparence ne doit pas mimer de la matière en pâte à modeler.

Champs à divergence nulle

Comme décrite au troisième point des possibilités de contraintes de volume, l'utilisation de champ de vecteur à divergence nulle permet de définir sur un plan théorique des méthodes qui préservent localement le volume de manière automatique.

Fluides incompressibles

La contrainte de non-divergence d'un champ de vitesse est connue dans le domaine de la mécanique des fluides sous le terme *conservation de masse* et traduit le principe d'incompressibilité de celui-ci.

Comme introduit en partie 1.1.2, les méthodes dites *level-set* [OS88, OF03] sont particulièrement adaptées à la simulation de surfaces à topologie variable telles que les fluides. L'utilisation d'un champ de vitesse à divergence nulle impose que la surface se déforme à volume constant. Cependant, la résolution numérique directe de l'équation aux dérivées partielles des *level-set* introduit une dissipation numérique liée à l'utilisation d'une grille cartésienne pour stocker les valeurs du potentiel implicite. Il s'en suit une perte de volume au cours de la déformation.

Des méthodes semi-Lagrangiennes sont ainsi mises en place afin de suivre explicitement certaines positions au cours de la déformation par intégration le long des lignes de flux et forcer la contrainte d'une préservation de volume plus précise [EFFM02].

Cependant, dans tous les cas, ces méthodes nécessitent la résolution numérique d'équations différentielles. Les erreurs numériques accumulées ne permettent que d'approximer la préservation du volume. De plus, l'expression du champ de vitesse est issue de la modélisation de la simulation, mais n'est pas manipulable par un artiste. Cette méthode ne permet donc pas de définir une déformation contrôlable.

Manipulation par champ de vecteur

Manipulation manuelle : Comme présentée en partie 1.3.1, Von Funck *et al.* [vFTS06, vFTS07] proposent une méthode de déformation comprenant les avantages de l'utilisation de champs de déplacement à divergence nulle tout en permettant leur manipulation interactive.

Cette approche permet la modélisation de formes à volume constant et sans auto-intersections tout comme dans le cas des sweepers [ACWK04b]. De manière identique, le contrôle de la déformation reste cependant peu adapté au cas d'animation de personnages. Il parait en effet fastidieux de définir pour chaque partie du personnage à déplacer un champ vectoriel dont l'influence et le type sont à redéfinir. De plus, les influences des outils sont définis de manière volumique (sphère, cylindre, ...) pour chaque partie à déformer ce qui est peu adapté au contrôle indépendant de différents membres d'un personnage (par exemple, la déformation d'un doigt peut déformer également son voisin spatiallement proche).

Application au skinning : Une possibilité d'utilisation de ces champs de vecteurs dans le cadre de l'animation consiste à rendre le choix de la primitive de déformation volumique automatique. C'est ce que propose Angelidis et Singh [AS07] en définissant une approche s'adaptant à l'animation par skinning.

Dans ce cas, chaque joint du squelette d'animation se voit générer dans l'espace un champ de déformation à divergence nulle dépendant de la rotation qui lui est associée. Les positions

des sommets du maillage sont alors intégrées le long des lignes de champs qui leur correspondent.

Les auteurs montrent que la déformation associée possède deux degrés de liberté suivant le choix de deux paramètres scalaires modifiant ainsi l'aspect finale de la forme comme illustré en fig. 1.48.

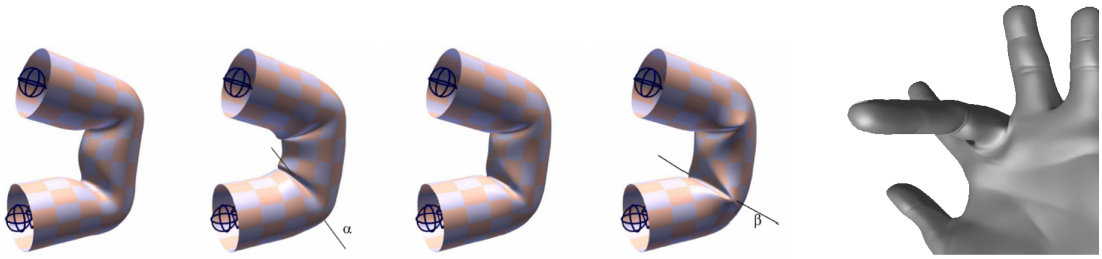


FIGURE 1.48 – *Illustration tirée de [AS07] d'un skinning à volume constant. Gauche : Différents effets obtenus pour la déformation d'un tube en fonction de deux paramètres scalaires de contrôles. Droite : Exemple d'application sur la déformation d'un doigt de main.*

Cette méthode possède cependant un certain nombre d'inconvénients :

- Les deux paramètres scalaires de modifications des champs ne permettent pas un contrôle direct et intuitif de la forme finale, mais simplement un ensemble d'essais-erreurs sur le paramétrage d'une fonction.
- L'intégration des sommets le long des lignes de champs est couteuse en temps de calcul et s'adapte mal à l'utilisation pour des maillages détaillés.
- La méthode globale nécessite une refonte complète de la méthode de skinning du fait de la nécessité d'intégration progressive au cours du temps et ne peut donc pas directement s'adapter aux pipe-line standard pré-existants des logiciels commerciaux.

Surfaces implicites à squelette

Dans le cas où une surface est représentée de manière implicite, Desbrun et Cani [DC95, CD97] mettent en place une méthode de contrôle de volume. 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 les frontières 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. Chaque instant, la fonction potentiel définissant la surface implicite est déformée suivant une équation différentielle permettant de limiter le changement de volume. 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.

1.4.3 Contraintes de longueurs

Certains objets tels que les vêtements ou le papier sont formés d'un enchevêtrement de fibres inextensibles. Au niveau macroscopique, ils possèdent la particularité de ne pouvoir ni s'étendre, ni se contracter. Certaines telles que les tissus préservent cette propriété de manière

approchée et non symétrique [13], alors que d'autres comme le papier préservent exactement cette propriété. Cette différence implique un traitement généralement séparé de ces deux problèmes. En effet, dans le cadre d'une application en graphique, le caractère plausible d'une animation de tissu tient avant tout à la présence de plis dynamiques au cours du mouvement (voir fig. 1,droite). La préservation numérique de toutes les distances entre couples de points n'étant pas le critère prédominant.

Au contraire, la déformation de papier nécessite une contrainte plus forte pour empêcher toute impression de compression ou d'extension. De plus, ce type d'objet est caractérisé par la présence de plis francs discontinus aux endroits soumis à une contrainte mécanique.
Nous noterons ainsi que les outils disponibles pour la mise en oeuvre d'animation de vêtements sont beaucoup plus développés que celle de l'animation de papier, les simulations numériques présentées en partie 1.2.4 n'étant pas adaptées à la modélisation de surfaces non lisses.

Nous abordons cette description des méthodes de préservation de longueur en cherchant des solutions aux problématiques suivantes :

- Localiser, modéliser et animer rapidement des vêtements présentant des plis. La présence de plis de vêtement étant le résultat d'une préservation approximée de longueur.
- Modéliser rapidement des surfaces non lisses telles que du papier froissé préservant des critères forts de préservation de longueur.

Définition et caractéristiques des isométries

Définition d'une isométrie

Les vêtements, tout comme le papier sont générés à partir de patron planaires. Sans perte de généralité, on pourra ainsi considérer la surface 3D comme étant issu d'un mapping f entre un domaine plan $\mathcal{D}$ appelé patron vers une surface 3D S.

$$f : \left\{ \begin{array}{ccc} \mathcal{D} \subset \mathbb{R}^2 & \rightarrow & S \subset \mathbb{R}^3 \\ \mathbf{q} = (x, y) & \mapsto & f(\mathbf{q}) \end{array} \right.$$

Par définition, f est une fonction isométrique ssi elle préserve les longueurs. L'isométrie peut se définir formellement et de manière générale par le fait que toute courbe de $\mathcal{D}$ soit transformée par f en une courbe de même longueur [Car76] (voir fig. 1.49). Soit,

$$f \in \text{isometry} \;\Leftrightarrow\; \forall \mathcal{C} \subset \mathcal{D}\ , \quad \int_u \|c'(u)\|\ \mathrm{d}u = \int_u \|(f \circ c)'(u)\|\ \mathrm{d}u\ . \tag{1.29}$$

$\mathcal{C}$ étant la courbe associée à la fonction paramétrique $c : u \in \mathbb{R} \mapsto c(u) \in \mathbb{R}^2$. Notons que dans le cas où f n'est pas dérivable, on pourra scinder le second membre en domaines différentiables par morceaux. Cette relation n'est cependant pas implémentable numériquement.

Notons que la préservation d'isométrie exacte est une relation forte impliquant la préservation des angles et des aires.

Dans le cas d'un maillage triangulaire [14], l'eq. (1.29) peut s'exprimer de manière équivalente par le respect des longueurs des arêtes des polygones.

Soit $\mathcal{E}$ l'ensemble des arêtes du maillage triangulaire, on a alors

$$f \in \text{isometry} \Leftrightarrow \forall (i, j) \in \mathcal{E}\ , \quad \|\mathbf{q}_i - \mathbf{q}_j\| = \|f(\mathbf{q}_i) - f(\mathbf{q}_j)\|\ , \tag{1.30}$$

13. L'extension étant réalisable, mais pas la contraction
14. Notons que ceci n'est pas pas applicable à des quadrangles où il faut ajouter une information d'angle

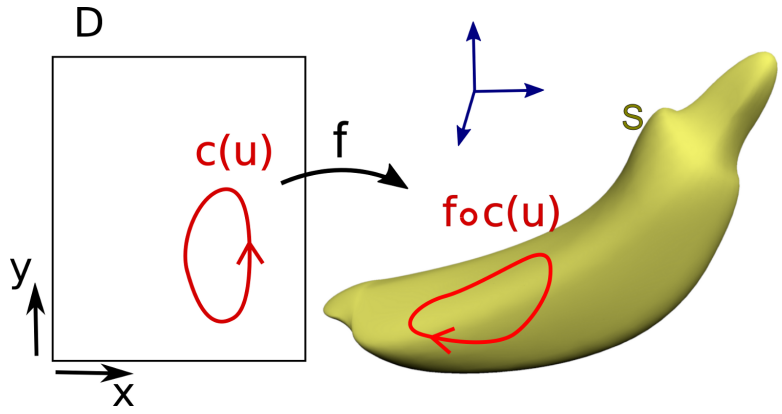


FIGURE 1.49 – *Transformation d'une courbe d'un domaine plan $\mathcal{D}$ sur une surface S par un mapping f. f est une isométrie si les courbes $2D$ et $3D$ ont mêmes longueurs.*

ou encore

$$E_L = \sum_{(i,j)\in\mathcal{E}} \left(\|\mathbf{q}_i - \mathbf{q}_j\| - \|f(\mathbf{q}_i) - f(\mathbf{q}_j)\|\right)^2 . \tag{1.31}$$

Le terme E_L est donc une mesure calculable de préservation globale de l'isométrie. Notons que ce type de mesure d'adapte directement aux cas des déformations basées sur les masses-ressorts (partie 1.2.3). Elle ne définit par contre pas une métrique locale de la déformation en isométrie, et de plus elle est fortement dépendante de la triangulation sous-jacente.

Supposons désormais que f soit différentiable. La première forme fondamentale de f [Kre91] est la matrice I de la forme quadratique représentant l'élongation ou la compression locale par rapport à la paramétrisation 2D.

$$\mathrm{I} = \begin{pmatrix} \left(\frac{\partial f}{\partial x}\right)^2 & \left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\rangle \\ \left\langle \frac{\partial f}{\partial x}, \frac{\partial x}{\partial y} \right\rangle & \left(\frac{\partial f}{\partial y}\right)^2 \end{pmatrix} .$$

Dans le cas où la surface d'origine est planaire, c'est à dire identique à la paramétrisation, cette matrice se ramène également au tenseur droit de Cauchy-Green de la mécanique des milieux continus [Spe04]. Nous obtenons alors une troisième définition de l'isométrie donnée par

$$f \in \text{isometry} \Leftrightarrow \forall \mathbf{q} \in \mathcal{D} \ , \ \ \mathrm{I}(\mathbf{q}) = \text{Identity} . \tag{1.32}$$

Cette mesure d'isométrie locale est calculable pour une surface différentiable paramétrique. Nous établirons en partie 3.1.1 une extension de ce calcul nous permettant d'exprimer ce tenseur métrique dans le cas d'un maillage triangulaire.

Surface développable

D'un point de vue géométrique, deux surfaces G^2 déformées isométriquement possèdent la même courbure Gaussienne. Les surfaces que l'on considère étant isométriques à des patrons planaires, elles ont donc une courbure Gaussienne nulle en tout point. La réciproque étant

vraie [Kre91], il s'agit de la définition d'une surface dite *développable*, c'est à dire qui peut se dérouler à plat sans étirements ni contractions.

Nous nous plaçons désormais dans le cas où f est au moins C^2 pour définir les caractéristiques de ces surfaces. L'ensemble des définitions suivantes sont équivalentes :

- S est une surface développable
- Il existe un mapping $f : \mathcal{D} \subset \mathbb{R}^2 \mapsto S \subset \mathbb{R}^3$ isométrique et C^2.
- Le mapping $f : \mathcal{D} \mapsto S$ peut s'écrire sous la forme :

$$\begin{cases} f(u,v) = g_1(u) + v\, g_2(u) \\ \det(g_1', g_2, g_2') = 0\,. \end{cases}$$

  C'est-à-dire que S est une surface réglée dont les normales sont constantes le long des règles.
- La courbure de Gauss est nulle en tout point.
  Corollaire : En rappelant que l'application de Gauss (*Gauss Map*) est définie la fonction qui associe tout point de la surface à son vecteur normal sur la sphère unitée, il est possible d'en déduire que l'application de Gauss de S est une courbe sur la sphère et non une surface comme pour une surface non développable.

Notons qu'à la différence de la description par isométrie, il est possible de savoir qu'une surface est développable sans pour autant connaitre explicitement son patron planaire.

Enfin, de par les critères forts nécessaires à la développabilité d'une surface C^2, il est possible de définir un nombre fini de formes de bases engendrant ce type de surface. Une surface développable lisse ne peut ainsi être formée uniquement que de portions de : plan, cylindres, cônes ou surface engendrée par les tangentes d'une courbe [15].

Dans le cas d'un maillage polygonal, la notion de courbure doit être définie dans un sens discret [GWDS08]. Appelons K_i la courbure de Gauss discrète associée à un sommet i. Celle-ci est donnée par le *défaut angulaire* à ce sommet par rapport à 2π, c'est à dire

$$K_i = \frac{2\pi - \sum_{j \in \mathcal{V}_i^\star} \theta_{ji}}{A_i^v}\,, \tag{1.33}$$

avec θ_{ji} l'angle de la face j au sommet i, A_j^v l'aire de la région de Voronoi [16] associée au sommet i vis-à-vis de son 1-voisinage , et $\mathcal{V}_i^\star$ sont les indices des faces contenant j [17].

Notons également que lorsqu'un maillage représente une approximation d'une surface C^2 développable, alors celui-ci peut être décrit entièrement sans aucun sommet intérieur. En effet, les arêtes pouvant se placer sur les règles de la surface, les sommets du bord et leur connectivité définissent entièrement cette surface [RSW+07].

À partir de ces différentes définitions de préservation d'isométrie ou de développabilité, différentes méthodes de modélisation et d'animation ont été proposées.

15. ou similairement l'enveloppe des plans osculateurs d'une courbe
16. On approxime souvent cette aire, par une pondération de l'aire des faces $1/3 \sum_{j \in \mathcal{V}_i^\star} A_j$, où A_j est l'aire de la face j.
17. Étoilé de j

Animation de plis de vêtements

Comme nous l'avons vu en partie 1.2.4, les approches physiques sont capables de réaliser des animations réalistes de vêtements plissés. Elles se heurtent à deux limitations principales :

- L'absence de contrôle direct de la forme finale obtenue à l'issue de l'intégration temporelle, mais passant par l'utilisation de paramètres de raideurs peu intuitifs.
- Le temps de calcul nécessaire à l'animation d'un maillage inextensible détaillé.

Compte tenu de la présence importante de vêtements dans le cinéma d'animation et les jeux vidéo, de nombreuses approches ont déjà été étudiées afin de modéliser des vêtements inextensibles présentant des plis. Nous distinguons trois approches principales : L'ajout de contraintes supplémentaires dans les simulations physiques numériques, les méthodes d'optimisation globale et les approches géométriques.

Ajout de contraintes dans la simulation

Provot [Pro97] est l'un des premiers à publier une méthode physique qui se veut à la fois rapide, et permettant de limiter les variations de longueur.
Pour cela, il introduit une limite d'extension des longueurs des arêtes d'un maillage masses-ressorts par une méthode de projection itérative géométrique des sommets. Le comportement non linéaire des ressorts est ainsi modélisé sans apporter de problèmes de divergence numérique pendant l'étape d'intégration. La méthode est simple à mettre en oeuvre, mais la convergence de la projection géométrique n'est pas optimale.



FIGURE 1.50 – *Simulation de tissu quasi inextensible. Image provenant de [GHF+07].*

Goldenthal *et al.* [GHF+07] améliorent l'approche de projection sur la contrainte de préservation de longueur telle que définie à l'eq. (1.30) en introduisant celle-ci dans l'optimisation globale à l'aide de la formulation par multiplicateurs de Lagrange. Une méthode de résolution spécifique appelée *Fast projection method* est proposée. L'approche permet de modéliser des maillages à faces quadrilatérales présentant des plis et une isométrie approchée (voir fig. 1.50). Des approches similaires implémentables sur des architectures parallèles furent également étudiées [BB08, SSBT08].

English et Bridson [EB08] étendent cette approche au cas de surfaces triangulaires. Ils suggèrent de modifier la représentation de la surface au sein même de l'animation physique. À la place de considérer un maillage dont les variables de positions se placent au sommet des faces, ils considèrent celles-ci positionnées au milieu des arêtes. Une telle représentation est dite *maillage non conforme*, car ils n'imposent plus la continuité des sommets. Cette représentation permet d'augmenter le nombre de degrés de liberté et évite le problème de *locking* discuté en partie 1.2.4.
Pour passer de l'étape t à $t + \Delta t$, les auteurs entremêlent alors :

- une étape d'intégration temporelle standard sur le maillage non conforme ainsi défini ;

– une étape de projection inspirée de [GHF+07] sur la contrainte de longueur pour les positions définies au milieu des arêtes. L'amélioration apportée concerne un schéma d'intégration plus précis, celui-ci étant d'ordre 2 par rapport à l'approximation d'ordre 1 de l'approche de Goldenthal *et al.*.

Les auteurs montrent que le gain de degré de liberté apporté par la structure non conforme permet de faciliter la génération de plis, mais rend en contrepartie la surface résultante non continue. Pour obtenir un modèle visuel convaincant ainsi que pour pouvoir traiter précisément les collisions, un second maillage continu est généré au cours des itérations à partir de cette surface non conforme par simple moyennage de la position des sommets.
Il en résulte ainsi une surface de tissu dynamique présentant de nombreux plis et limitant approximativement les déformations de longueur.

Enfin similairement à notre approche, l'ajout de détails géométriques de plis peut s'ajouter sur un maillage à basse résolution. Kang *et al.* [KCCL01] génèrent des plis en introduisant une variation sinusoïdale sur des carreaux splines. Larboulette et Cani [LC04] définissent des déformations similaires à partir de courbes paramétriques de plis plus aisées à contrôler et ne nécessitant pas de carreaux tensoriels. La préservation de longueur n'est cependant pas garantie. Sauvage *et al.* [SHB06] définissent quant à eut une approche multiresolution préservant la longueur de la courbe.
Finalement, Muller *et al.* [MC10] proposent récemment une méthode capable d'ajouter automatiquement des détails de plis par l'introduction d'un second solveur venant déplacer les sommets du maillage subdivisé afin que la contrainte de longueur soit mieux préservée.

D'un autre côté, certains auteurs se basent sur une modélisation de la déformation issue de la mécanique des milieux continus en se basant sur la minimisation de l'élongation locale.
Les méthodes dites de *strain limiting* se basent sur la définition de l'eq. (1.32) et tentent de limiter l'écart entre le tenseur d'élongation I et l'identité.
Initiée par Bridson *et al.* [BFA02], l'idée est étendue par Thomaszewski *et al.* [TPS09] dans une formulation par éléments finis. Enfin récemment, Wang *et al.* [WOR10] montrent que cette approche peut également s'appliquer pour des déformations quasi-isométriques d'objets quelconques.
Ce type d'approche permet d'obtenir des résultats d'un grand réalisme, mais reste excessivement couteux (plusieurs minutes pas seconde de simulation).

Optimisation globale

D'autres approches facilitent la génération de formes de tissus plissés en se basant sur une optimisation de fonctionnelle d'énergie quantifiant la développabilité. Bien que non basée sur des équations physiques comme les papiers précédents, une telle formulation peut permettre d'obtenir des résultats visuels plausibles tout en permettant de contrôler plus aisément le résultat final, par exemple en ajoutant des contraintes spatio-temporelles de positions.

En partant de la définition de l'eq. (1.33), Wang et Tang [WT04] proposent de modéliser une surface S_2 quasiment développable en déformant une surface S_1 donnée. Pour cela, il proposent de minimiser une énergie E_1 basée sur l'erreur de courbure Gaussienne

$$E_1 = \sum_{i=1}^{N_s} \left(2\pi - \sum_{j \in \mathcal{V}_i^\star} \theta_{ij} \right)^2 . \tag{1.34}$$

De plus, afin de minimiser l'écart à la surface S_1, ils introduisent une seconde énergie quantifiant l'élongation des arêtes

$$E_2 = \sum_{(i,j)\in\mathcal{E}} \left(\|\mathbf{p}_i^{S_1} - \mathbf{p}_j^{S_1}\| - \|\mathbf{p}_i^{S_2} - \mathbf{p}_j^{S_2}\| \right)^2 ,$$

avec $\mathbf{p}^{S_1}$ et $\mathbf{p}^{S_2}$ référant aux positions des sommets de S_1 et S_2, les indices des arêtes étant données par $\mathcal{E}$.

L'énergie totale à minimiser est alors définie par E, avec

$$E = \eta E_1 + (1-\eta) E_2 , \tag{1.35}$$

η étant un facteur de pondération scalaire $\eta \in [0,1]$.

La résolution de ce système non linéaire global est réalisée par une méthode standard de gradient conjugué [QSS00]. Pour réduire le coût en temps de calcul (plusieurs minutes pour un maillage de 3000 sommets), les auteurs proposent également une solution approchée locale. Dans ce cas, chaque sommet est déplacé le long de la normale à la surface de manière à satisfaire la contrainte de développabilitée. Cette procédure d'optimisation gloutonne est alors poursuivie itérativement sur chaque sommet classé prioritairement suivant leur erreur angulaire.

Wang [Wan08] étend l'approche au cas des surfaces de subdivision afin de définir un algorithme convergeant vers une surface développable au cours de la subdivision. Ils utilisent dans un premier temps une estimation d'une surface minimale à partir d'une minimisation de l'opérateur Laplacien (voir partie 1.3.3), puis optimisent localement la développabilité de celle-ci suivant une approche similaire à celle décrite précédemment.

Sur une approche similaire, Tang et Chen [TC09, CT10] ajoutent des contraintes de positions sur certains sommets de la surface finale, ainsi que l'action de la gravité. Ils parviennent ainsi à définir une modélisation de surface quasi-développable dont l'aspect général peut être contrôlé manuellement, et dont les plis se localisent de manière automatique (voir fig. 1.51).

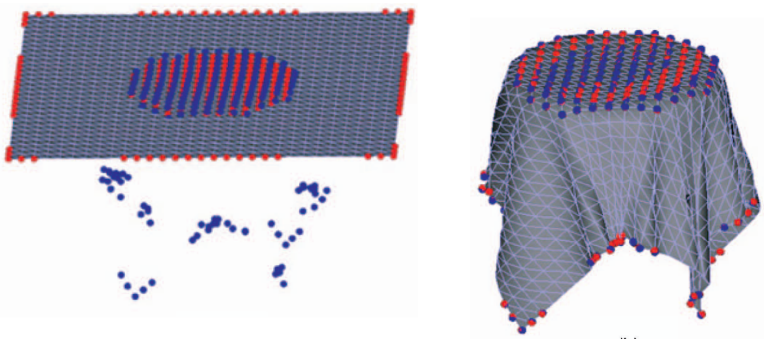


FIGURE 1.51 – *Déformation d'une surface développable. La figure de gauche représente le maillage initial et les contraintes positionnelles souhaitées. La figure de droite représente le résultat obtenu. Image provenant de [TC09].*

Dans un autre ordre d'idée, Popa *et al.* [PZB+09] utilisent une contrainte spatio-temporelle afin de permettre l'animation de telles surfaces au cours du temps. La méthode s'applique dans le cas où l'entrée est une capture vidéo de vêtement animé, et dont la résolution ne permet pas

de distinguer les plis. Ces derniers doivent être localisés et placés automatiquement. La minimisation a cette fois lieu sur le critère de développabilité (voir eq. (1.34)) et sur la contrainte pondérée d'orientation des normales en fonction de l'analyse de l'entrée vidéo. Cette approche d'optimisation reste cependant excessivement lourde en temps de calcul et nécessite une vidéo en entrée.

L'ensemble de ces méthodes permet d'apporter un degré de contrôle artistique supplémentaire sur la forme générale de la surface par rapport à une approche de simulation pure. Cependant, elles sont généralement hors de portée d'une manipulation en temps interactif. De plus, il reste difficile d'obtenir un contrôle direct sur l'aspect final des plis engendrés. Enfin, la taille du maillage étant fixée initialement, il n'est pas possible d'engendrer localement des plis sans augmenter de manière significative la taille des systèmes d'équations à résoudre (exprimées globalement sur tout le maillage), ce qui pose des problèmes de temps de calcul et de coût mémoire.

Approches géométriques

Afin de limiter le coût de calcul de la génération de plis sur du tissu, certaines méthodes, similairement aux nôtres, proposent de modéliser ces détails à partir d'information géométriques.

Formes géométriques prédéfinies Decaudin *et al.* [DJW+06] introduisent la connaissance a-priori du comportement des plis de tissus lorsque celui-ci est enroulé autour d'une forme cylindrique, telle qu'un bras ou un tronc de personnage. Ceux-ci sont alors précédéralement appliqués sur le maillage par une déformation géométrique connue explicitement. Pour modéliser différents types de tissus, il est possible de contrôler directement la densité des plis formés.

Bien qu'applicable en temps réel et parfaitement contrôlable dans son domaine de déformation, la méthode dépend largement d'une paramétrisation spécifique du maillage adaptée à ces plis, et de l'application d'un nombre limité de déformations (compression, torsion) ne pouvant être combinées avec une animation de vêtement quelconque.

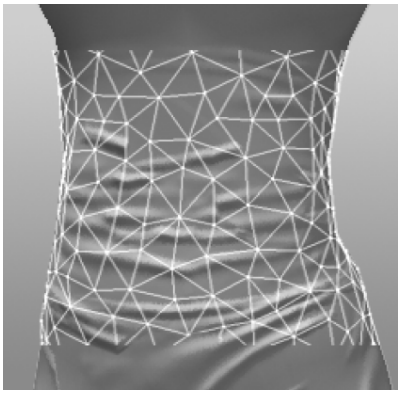


FIGURE 1.52 – *Carte de déformation de normales appliqué sur un maillage grossier. Image provenant de [HBVMT99].*

Textures de plis Afin de modéliser rapidement des plis de formes quelconques indépendamment du maillage original, plusieurs approches se basent sur l'intégration des plis au niveau de la texture.

Hadap *et al.* [HBVMT99] calculent automatiquement une carte d'intensité de plis sur la surface. Cette carte vient alors pondérer une texture pré-calculée modélisant la forme des plis (voir fig. 1.52). La pondération est calculée de manière à ce que l'aire des triangles compressés au cours de la déformation soit approximativement restaurée. La texture pondérée résultante est ensuite utilisée en tant que carte de normales (bump-map) permettant ainsi son utilisation dans un contexte

temps-réel en donnant l'impression d'un tissu plissé. L'approche peut également s'appliquer à de l'animation, mais nécessite une interpolation sur des textures de plis pré-calculées. La définition de ces textures se révèle ainsi fastidieuse, et de plus l'interpolation de texture ne préserve pas l'aspect de plis à l'orientation marquée.

Kimmerle *et al.* [KWH04] étendent cette idée pour générer procéduralement les formes des plis qui, similairement à notre application, s'orientent automatiquement suivant la direction de compression principale. Le cas d'animation peut ainsi être plus aisément mis en place. Les formes procédurales des plis se résument cependant à des oscillations sinusoïdales dans une direction particulière qui ne peuvent modéliser qu'une gamme limitée de type de plis. De plus, le rapprochement voir le mélange de plis ayant des orientations différentes implique des interférences visuellement non réalistes.

Courbes de plis manuelles Les approches procédurales précédentes permettent de définir automatiquement l'amplitude des plis en fonction de la compression exercée afin de rendre la surface plus développable. Cependant, la localisation de ces plis reste difficile à gérer à partir de cartes de textures peintes ou générées par sinusoides.
Afin de donner un maximum de liberté à l'artiste manipulant le modèle de vêtement, Cutler *et al.* [CGW+05] définissent une méthode de modélisation manuelle. Les plis sont alors définis comme une déformation locale ayant lieu le long d'une courbe vivant sur la surface comme illustrée en fig. 1.53. La courbe et le rayon d'influence potentiel de la déformation sont définis manuellement par l'artiste pour un certain nombre de positions clés. Les déformations sont alors interpolées au cours de l'animation et peuvent se calculer interactivement.



FIGURE 1.53 – *Courbes de plis dessinés manuellement sur une surface.L'utilisateur peut paramétrer la localisation et le rayon d'influence de ces courbes au cours du temps. Image provenant de [CGW+05].*

Cette approche permet un grand contrôle artiste adapté à la création de plis, il requiert cependant une expertise pour localiser et animer de tels plis de manière réaliste au cours du temps. De plus, l'approche demande un investissement très important en temps de modélisation manuelle. Enfin, l'interpolation entre les déformations de plis ayant des orientations différentes génère un comportement d'apparition et de disparition peu réaliste.
Notre approche de génération de plis de vêtement utilise le même encodage des plis sous forme de courbes localisant leur position, et de rayon permettant de définir leur amplitude. Cependant, nous définirons une méthode automatique permettant de générer ces courbes à la volée et lors

d'animation sur les régions compressées du maillage originale. De plus, nous définirons une méthode de déformation induite par ces courbes permettant de réaliser des mélanges naturels de plis.

Après avoir étudié le cas particulier de méthodes permettant de générer des plis de vêtements à partir d'une contrainte approchée de préservation de longueur, nous étudions le cas particulier de construction de surfaces développables. En particulier, nous souhaitons pouvoir générer des formes se rapprochant de papier froissé.

Modélisation et déformation de surfaces exactement développables

Nous rapellons que nous intéresserons spécifiquement à la modélisation du papier en partie 4. D'un point de vue microscopique, le papier se compose d'un assemblage en couche de fibres de bois entremellées (voir fig. 1.54). L'épaisseur d'une feuille de papier étant très faible devant les dimensions de la feuille ($\simeq$ 0.1mm), celle-ci est généralement négligée pour son apparence macroscopique. Néanmoins les caractéristiques de ces fibres sont responsables des propriétées géométrique visible à notre échelle.



FIGURE 1.54 – *Coupe microscopique d'un morceau de papier obtenue par micro-tomographies (rayons-X) illustrant l'enchevêtrement de fibres. Image issue de [AN06].*

D'une manière synthétique, nous pourrons retenir que
- L'assemblage de fibres accolées ne permet pas les élongations et les contractions de la surface de papier.
- Soumis à des contraintes importantes, les fibres se brisent de manière irréverssible. On parle alors de comportement *plastique*. Géométriquement, ce comportement se traduit par l'apparition de plis francs de manière permanente sur la surface.

Les approches de simulation physique intègrent classiquement des propriétés différentielles de surfaces continues de manière approchée. Elles sont donc peu adaptées à la simulation de surfaces présentant des plis francs tels que du papier, ou à la recherche d'une préservation exacte de propriétés géométriques comme la développabilité.

Seules quelques approches [GHDS03, BGW06] ont été proposées pour permettre la simulation physique de type plaque mince s'adaptant à une surface présentant des discontinuités de tangentes. Cependant, les arêtes vives doivent être explicitement définies manuellement par l'utilisateur au préalable (voir fig. 1.55). Kang *et al.* [KZC09] proposent une méthode de subdivision d'arête adaptative d'un système masses-ressorts permettant de prendre en compte les brisures nettes d'une surface modélisant du papier. La méthode permet l'animation rapide de telles surfaces, mais les lignes de brisures sont contraintes d'agir le long des arêtes du maillage original ce qui introduit des artefacts visuels.



FIGURE 1.55 – *[GHDS03] propose une simulation de papier réaliste mais nécessitant un travail manuel de définition d'arêtes vives.*

En parallèle, des approches géométriques tentent de modéliser spécifiquement des surfaces développables. De part son intérêt principal en CAO, typiquement pour le design d'objets manufacturé ou l'architecture, les surfaces développables furent tout d'abord étudiées par la création de patchs paramétriques obtenus par produit tensoriel [18] [PF95, CM98, CS02, Aum03, PS07]. Cela restreint cependant leur utilisation à des modèles se prêtant à une représentation par patchs tensoriels et le raccord entre patchs reste complexe à gérer.

Par la suite, Frey [Fre02, Fre04] propose une approche algorithmique pour trianguler un bord 3D le long des règles d'une surface développable. En considérant une surface C^2 développable, l'auteur cherche une manière de trianguler une courbe discrète représentant le bord d'une surface développable sans ajout de points intérieurs.
Pour cela, il se base sur la définition réglée de la surface développable. Ils remarquent notamment que le fait que les normales aux règles se placent sur un plan tangent constant implique que celles-ci se situent entièrement sur le bord du polygone convexe englobant cette règle, et non à l'intérieur de celui-ci. Frey propose alors de construire cette triangulation en cherchant à lier les sommets tel que les arêtes se placent sur l'enveloppe convexe formé par le bord 3D.
La méthode est cependant restreinte au cas de bords 3D pouvant se projeter sans ambiguïté le long d'un axe. L'approche est étendue par Rose *et al.* [RSW+07] sans le cas d'un contour fermé 3D quelconque issu de croquis (voir fig. 1.56).

Si les données d'entrée correspondent à un ensemble de positions non structurée de l'espace [19], Peternell *et al.* [Pet04] proposent une approximation par une surface développable Bo et Wang [BW07] proposent quant à eux de modéliser une telle surface à partir de la donnée de courbes géodésiques sur celle-ci.
Liu *et al.* [LPW+06] proposent une méthode de subdivision de maillage quadrilatère convergeant vers des portions de cônes développables dans le cadre d'une utilisation architecturale.

18. La littérature du domaine est étendue, mais celle-ci ne résolvant pas les problématiques centrales à cette thèse, nous ne la présenterons pas de manière exhaustive
19. Comme par exemple, des mesures issues de scanners.

Enfin, Perriollat [Per08] développe la modélisation de surface développables lisses dans le cas de données d'entrées issues de photos réelles.

Toutes ces approches permettent de converger vers des surfaces développables. Cependant, elles ne prennent pas en compte la préservation d'isométrie par rapport à un patron qui peut être connu. De plus, elles ne modélisent que des portions de surfaces C^2 développables, c'est à dire des surfaces réglées sans points singuliers et sans arêtes vives. Ces surfaces restent donc contraintes à rester dans le domaine défini par le polygone convexe s'appuyant sur leur bord initial. Le cas du papier froissé ne pourrait donc pas être traité par ces approches sans un découpage préalable le long d'arêtes vives qui seraient à définir manuellement.

De même, l'ensemble des approches basées sur une optimisation globale présentée précédemment [WT04, Wan08, TC09, PZB+09, CT10] peuvent traiter la contrainte d'isométrie, mais nécessitent la connaissance préalable d'un maillage fixé. Celui-ci ne permet que difficilement la prise en compte de bords francs qui ne seraient pas alignés avec les arêtes du maillage initial.

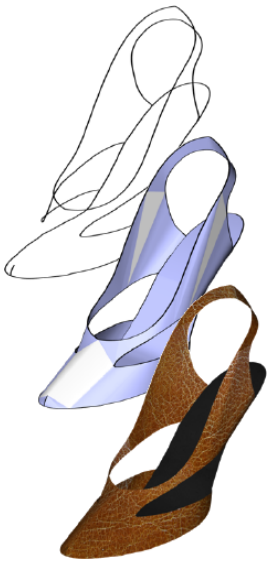
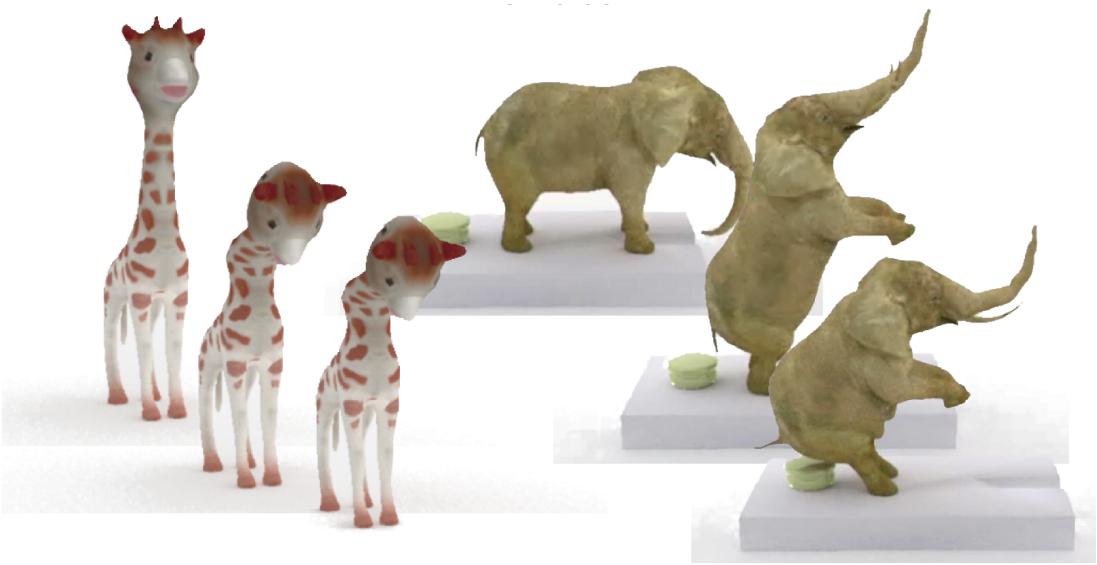


FIGURE 1.56 – *Surface développable lisse interpolant des bords 3D issus d'un croquis. Image provenant de [RSW+07].*

CHAPITRE

2

DÉFORMATION DE MAILLAGES HABILLANT UN SQUELETTE. APPLICATION À L'ANIMATION DE CRÉATURES VIRTUELLES.



LE besoin de créatures virtuelles animées devient de plus en plus important dans les scènes virtuelles telles que le jeu vidéo ou le cinéma d'animation. Il est fréquent d'avoir besoin de définir des créatures fantastiques, dont l'animation n'obéit pas aux lois physiques classiques. Il est donc nécessaire de contrôler ces créatures par des méthodes de déformations procédurales qui soient à la fois plausible visuellement, dont le contrôle soit intuitif, et dont le calcul soit suffisamment rapide pour obtenir un retour visuel interactif.

Le skinning présenté en partie 1.3.1 est l'une des méthodes de déformations et d'animation d'objets articulés la plus utilisée du milieu professionnel. Le pipe-line standard attaché à cette méthode est illustré en fig. 2.1. Elle permet à la fois la manipulation intuitive du squelette et un calcul rapide. La méthode possède cependant un défaut majeur : une déformation peu réaliste due à une perte de volume local lors de l'utilisation de rotations importantes.

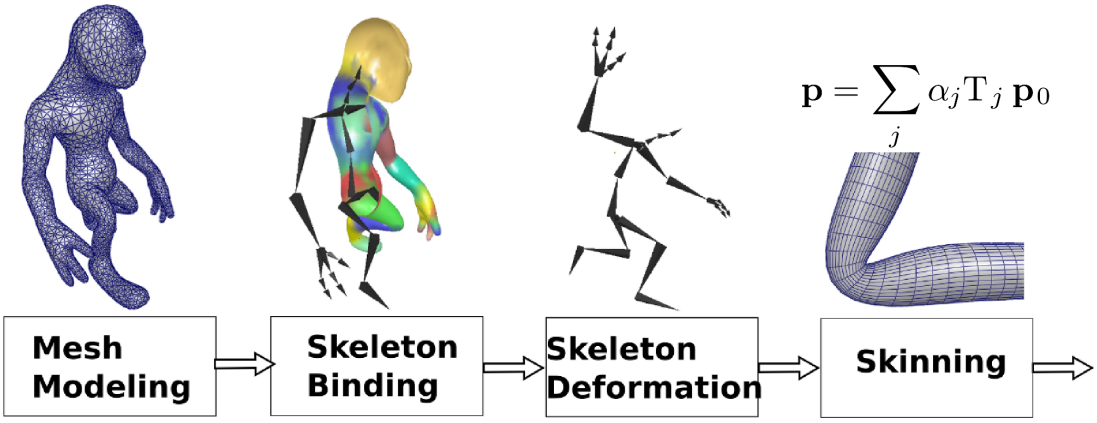


FIGURE 2.1 – *Pipe line standard de la déformation par skinning. Après l'étape de modélisation dans la pose de repos, un squelette est attaché à la surface à l'aide des poids de skinning : il s'agit de l'étape dite de Binding. Le squelette peut alors être animé et le maillage est déformé à l'aide de l'eq. (1.17).*

Les artistes numériques doivent ainsi mettre en place des outils de déformation secondaires pour limiter la perte de volume. Ils doivent de plus développer une expertise technique pour déterminer si ce défaut est lié intrinsèquement à la méthode, ou si celui peut être résolu par une modification des poids d'interpolation.

Nous proposons trois contributions pour faciliter l'animation de personnages et créatures virtuelles skinnées dans cette partie.

1. Le calcul automatique de poids de skinning permettant à un utilisateur non expert d'appliquer une déformation par skinning d'après l'unique information du maillage initial.
2. Une méthode d'estimation efficace du changement de volume local d'un modèle 3D animé.
3. Une préservation de volume locale et contrôlable permettant d'améliorer le réalisme d'une animation skinnée. Le pipe-line général de cette correction contrôlable est illustré en fig. 2.2.

Nous montrons finalement l'application de notre méthode sur différents cas concrets de créatures virtuelles et proposons l'intégration de cette méthode dans le cas d'une application préexistante de réalité augmentée. Enfin, nous montrons que cette méthode peut s'étendre à des contraintes supplémentaires tels que la préservation de l'aire.

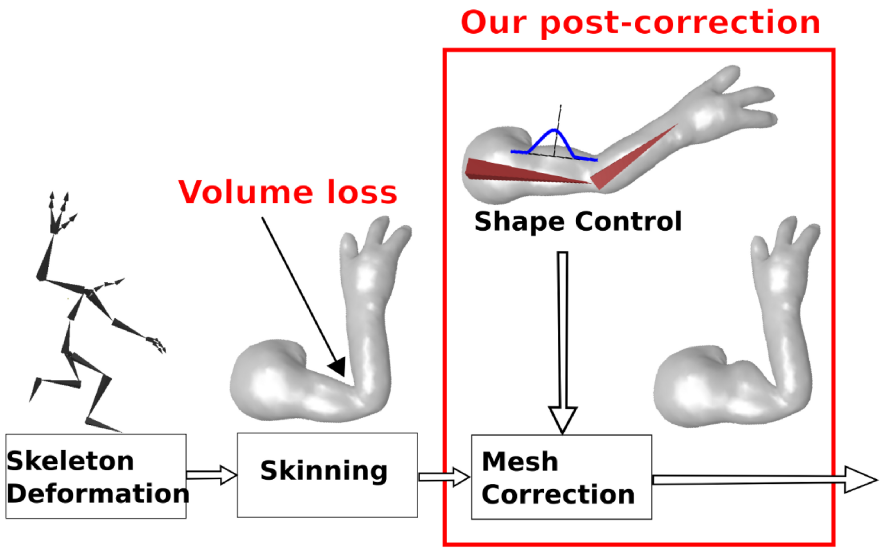


FIGURE 2.2 – *Pipe-line de notre approche de correction de volume. La déformation par skinning standard génère une perte locale de volume. Lors d'une étape de post-correction, nous compensons cette perte locale de volume de manière contrôlable.*

Les contributions principales de ce chapitre ont été présentées dans les publications suivantes : [RHC08a], [RHC08b], [RHC09], [RHC10].

2.1 CALCUL AUTOMATIQUE DE POIDS DE SKINNING

La déformation par skinning nécessite trois données d'entrée : Un maillage représentant la surface, un squelette d'animation, et les poids de skinning définissant l'influence des transformations du squelette d'animation par rapport aux sommets du maillage.

Les maillages sont généralement soit générés en entrée par un artiste, ou peuvent être issus de données numérisées dans le cas d'objets réels. Le squelette d'animation est quant à lui aisément modélisable à l'aide d'outils d'édition 3D tels que Blender, Maya, …. Dans le cas d'animaux bi-ou quadrupèdes, des méthodes de génération automatique de squelettes d'animations proches de squelette anatomiques peuvent également être utilisées comme présenté par Aujay *et al.* [AHLD07]. Enfin, les poids de skinning nécessitent généralement une attention toute particulière, car leur manipulation est moins intuitive. Les méthodes de définition de ces poids présentées en partie 1.3.1 montrent qu'il n'existe pas d'approche permettant de définir ceux-ci de manière automatique, rapide et robuste pour tout type de formes.

Si les outils de type *pinceaux* permettent des corrections manuelles indispensables, nous pensons qu'une définition automatique adéquate de poids de skinning en tant que point de départ permet d'épargner un travail fastidieux.

Nous proposons pour cela deux extensions des méthodes de définition automatique. La première étant une variante rapide de [BP07], alors que la seconde consiste à définir les poids de manière très robuste suivant une approche volumique.

2.1.1 Estimation rapide de poids de skinning à partir de la normale à la surface

Nous rappelons que la méthode de Baran *et al.* [BP07] définit les poids de skinning automatiquement grâce à deux étapes. L'étape 1 consiste à définir une distance cartésienne modifiée telle que l'influence d'un os n'est prise en compte qu'à condition que le sommet puisse être lié au point le plus proche de celui-ci par un segment n'intersectant pas la forme. L'étape 2 consiste à lisser les transitions des poids à l'aide d'une diffusion sur la surface. Nous proposons dans la suite une amélioration de l'étape 1 de cette approche.

Le calcul de cette étape nécessite $N_v \times N_j$ évaluations d'un lancé de rayons permettant de détecter la présence de l'intersection. Pour un maillage de l'ordre de $N_v = 10k$ sommets et présentant $N_j = 10$ os, cette évaluation de distance peut prendre une dizaine de secondes.

Lorsqu'une estimation interactive est nécessaire (ex. re-calcul dynamique des poids de skinning au cours d'une animation comme présentée en partie 2.6 et 2.7), il peut être avantageux de pouvoir rendre ce calcul moins couteux lorsque cela est possible.

Premièrement, nous savons que l'estimation du calcul du poids de skinning se réalise toujours sur un sommet de la surface. Nous avons donc en permanence accès à l'orientation sortante de la surface par l'information de la normale $\mathbf{n}_i$ en ce sommet i.
De plus, dans le cas de maillages suffisamment *simples* il est possible de considérer que ceux-ci sont formés d'assemblages de parties cylindriques (ex. bras, tronc, jambes, ...). Le calcul d'intersection par lancé de rayon, qui a pour but de détecter si l'os est situé dans la même partie cylindrique que le sommet, peut alors s'estimer rapidement par une comparaison entre la direction de ce rayon et la normale au sommet.
Soit $\mathbf{r}$ le vecteur liant le sommet courant au point le plus proche de l'os considéré (voir fig. 2.3).

- Si $\mathbf{r} \cdot \mathbf{n} < 0$ le rayon est dirigé vers l'intérieur du cylindre, on considère cet os comme influent.
- Si $\mathbf{r} \cdot \mathbf{n} > 0$ le rayon est dirigé vers l'extérieur du cylindre, il y a forcément intersection avec la surface, cet os est considéré comme ayant une influence nulle.

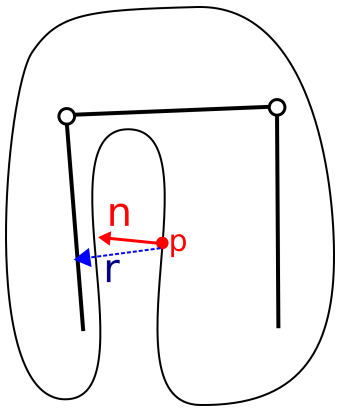


FIGURE 2.3 – *Evaluation rapide des poids de skinning par comparaison entre la direction normale au point courant* $\mathbf{n}$ *et le vecteur joignant le point le plus proche d'un os donné* $\mathbf{r}$.

Cette approche est suffisamment rapide pour permettre la mise à jour des poids de skinning de manière interactive. La mise à jour de tels poids pour un maillage de 10^4 sommets et 10 os se réalise en moins de $0.1s$.

L'approche se base cependant sur de nombreux a prioris (formes cylindriques de rayons approximativement homogènes, ...). Si on considère des cylindres de rayons fins et rapprochés, il devient nécessaire de réaliser le calcul d'intersection complet, car la supposition d'influence négligeable dans le cas de mauvaise interprétation dans le demi-espace $\mathbf{r} \cdot \mathbf{n} < 0$ n'est plus forcément réalisée.
De plus, si la surface présente de larges concavités telles que des replis de peaux, les calculs basés sur les intersections [BP07] ou la visibilité [WL08] ne sont plus à même de définir des influences significatives.

2.1.2 Calcul robuste des poids de skinning à partir d'une distance géodésique volumique

L'étape de diffusion de l'approche de Baran *et al.* [BP07] s'avère être très robuste dans le cas d'une géométrie complexe. Elle nécessite cependant d'être mise en place sur une segmentation initiale bien adaptée à la surface. Lorsque cet appariement initial est réalisé à partir d'une distance cartésienne modifiée, celle-ci ne peut pas gérer de manière satisfaisante de larges concavités.

Nous proposons de générer cet appariement, vu comme des poids de skinning rigide, à partir d'une information de distance géodésique volumique. Les avantages retenus étant que cette distance considérera de manière robuste la géométrie de la surface indépendamment de sa complexité.

Principe de la propagation de distance

Considérons un modèle voxelisé de la surface comme illustré en fig. 2.5.
Pour construire une telle voxelisation, la surface est plongée dans une grille volumique régulière. Les voxels touchants touchant les triangles de la surface sont tout d'abord remplis. La surface est donc définie par un ensemble connexe de voxels. Nous utilisons ensuite un algorithme de type *flood fill* à partir d'une graine définie à l'intérieur du volume. Au final, nous obtenons un ensemble de voxels remplissant l'intérieur de la forme et englobant la surface comme illustré en fig. 2.4.

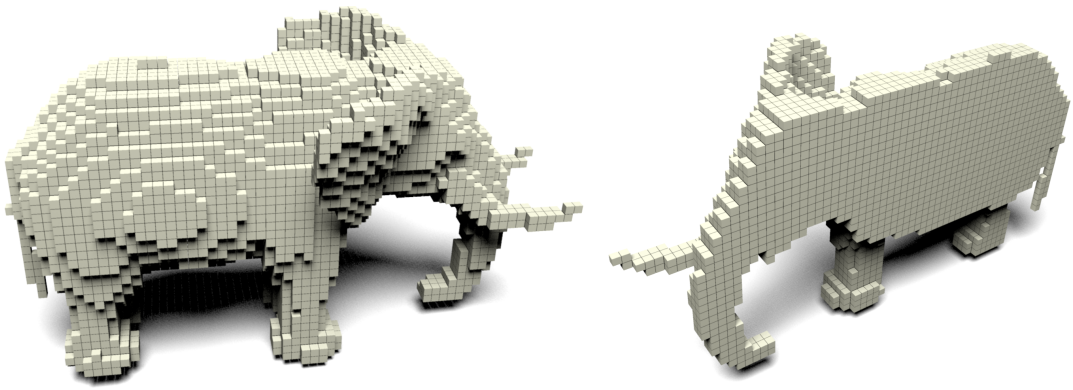


FIGURE 2.4 – *Exemple de voxelisation de la surface de l'éléphant. Le bord et l'intérieur de la forme sont remplis.*

Notons que la taille des voxels doit être sélectionnée pour être suffisamment fine afin de ne pas connecter arbitrairement les régions géométriquement non liées.

Dans une seconde étape, nous générons une carte de distance volumique associée à chaque os. Pour cela, nous mettons en place une propagation de distance de proche en proche suivant un algorithme de Dijkstra [Gib99] où les noeuds du graphe sont les voxels.
La distance entre les voxels situés sur les os étant considérée comme nulle, seuls les autres voxels situés à l'intérieur du volume ou sur la surface sont considérés lors de la propagation.

Une fois les N_j cartes volumique de distance calculée, les sommets de la surface sont attachés à l'os dont la distance correspondante est la plus faible. C'est-à-dire qu'un poids de 1 est associé à l'os le plus proche, et 0 pour les autres.

Enfin, similairement à [BP07] nous réalisons une étape de lissage par diffusion surfacique permettant de réaliser des transitions douces entre les différents segments. Les étapes de cet algorithme sont illustrées en fig. 2.5

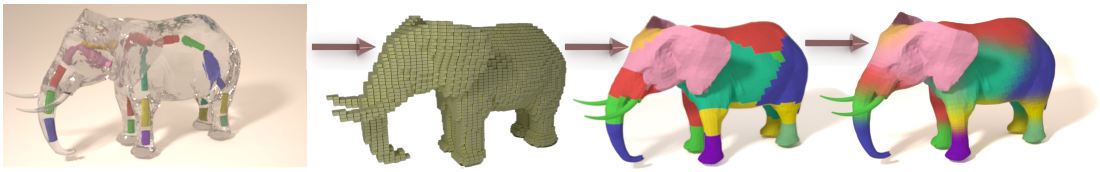


FIGURE 2.5 – *Calcul automatique de poids de skinning par distance géodésique. De gauche à droite : Modèle original et son squelette ; Modèle voxelisé ; segmentation suivant l'os le plus proche géodésiquement ; lissage des poids de skinning par diffusion.*

Résultats

Ce type d'approche permet la définition automatique de poids de skinning s'adaptant à des maillages complexes.
Un exemple de maillage illustrant une tête détaillée de dragon est proposé en fig. 2.6. L'utilisation d'une distance cartésienne modifiée ou basée sur un calcul de visibilité viendrait associer la déformation de la langue à l'os du fond de la gorge, et non à celle de la mâchoire. Le fait de se baser sur une propagation de distance volumique permet de résoudre ce problème et d'obtenir les appariements souhaités lors de l'articulation des os.

Cette approche est spécialement adaptée pour traiter le cas de géométrie présentant des concavités. Elle est cependant couteuse à mettre en place à la fois en terme de mémoire vis-à-vis du stockage des cartes de distances volumique, ainsi que du temps de calcul. Pour une carte de 256^3 voxels, nous obtenons un calcul permettant de traiter un os en $\simeq$20 secondes. Un maillage complet possédant 10 à 20 os sera donc traité non interactivement en 5 à 10 min.

Synthèse sur la génération de poids de skinning

Nous avons vu qu'il était possible, à partir de la donnée d'un maillage initial de définir de manière automatique des poids de skinning permettant de manipuler un personnage ou une créature virtuelle. Nous avons pour cela présenté deux nouvelles approches permettant de soit :

– Définir rapidement des poids dans le cas d'une géométrie simple.

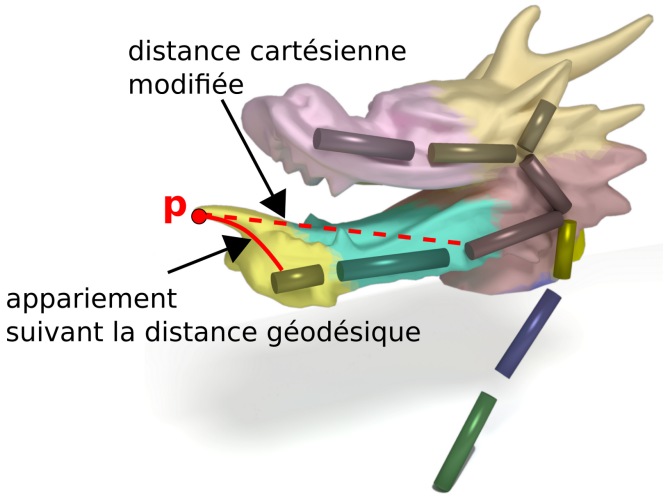


FIGURE 2.6 – *Appariement des sommets pour une forme complexe largement concave. Les sommets de la langue sont correctement affectés à l'os de l'extrémité de la mâchoire dans le cas de l'utilisation d'une distance géodésique volumique. Dans le cas d'une distance cartésienne modifiée par calculs d'intersection, l'appariement attache la langue à un os plus lointain.*

– Définir de manière robuste des poids de skinning dans le cas d'une géométrie complexe.

Dans la suite de ce chapitre, nous supposons ces poids connus et considérons le cas de la déformation proprement dite. En particulier, nous nous intéressons à la préservation, ou au contrôle, du volume de l'objet déformé.

2.2 CALCUL EFFICACE DU CHANGEMENT DE VOLUME LOCAL D'OBJETS ANIMÉS

Dans un premier temps, nous décrivons des approches permettant le calcul exact ou approché de la modification du volume d'un objet animé. Contrairement aux approches précédentes n'assurant qu'un contrôle global du volume [HML99, SZT+07, SHB07], l'approche présentée dans ce manuscrit estime celui-ci localement. Disposer d'une estimation locale est de première importance : en effet la perte de volume localisée en un endroit particulier d'un personnage (ex. bras) ne doit généralement pas impacter d'autres régions éloignées (ex. d'autres parties du corps autre que le bras). Il est donc nécessaire de localiser de manière automatique ces modifications de volume.

Mise en place du problème

Soient une surface initiale $\overline{S}$ et une surface déformée S. On pourra considérer le cas particulier où la surface S est obtenue par la déformation de type *skinning* de la pose de repos $\overline{S}$. Cependant, S peut être obtenue de manière générale suivant une déformation quelconque et inconnue d'une surface $\overline{S}$.
En notant V_S (resp. $V_{\overline{S}}$) le volume du domaine délimité par la surface S (resp. $\overline{S}$), on cherche à estimer le changement de volume $\Delta V = V_S - V_{\overline{S}}$ localement pour chaque partie de l'objet comme illustré en fig. 2.7.

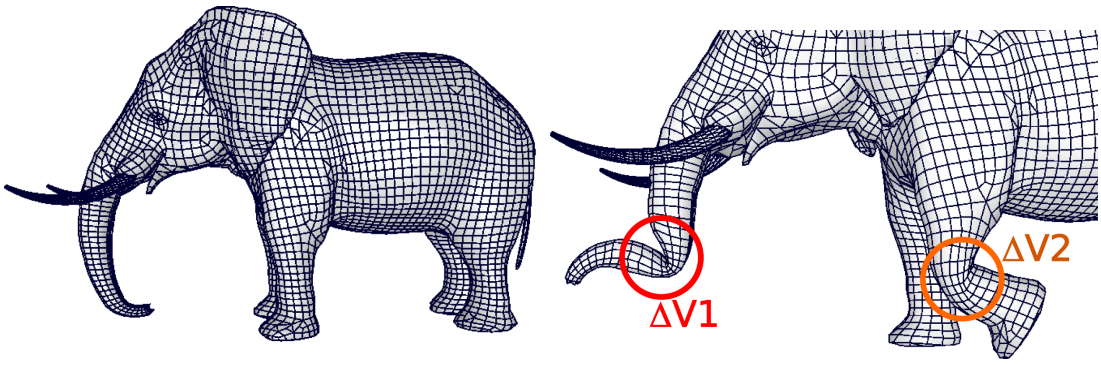


FIGURE 2.7 – *Illustration d'un changement local de volume. Gauche : Maillage original. Droite : Déformation importante par skinning de la trompe et de la patte avant avec perte de volume visible. Les zones de pertes de volumes locales sont entourées et associées à une valeur ΔV_1 et ΔV_2. Ces deux valeurs doivent être connues indépendamment l'une de l'autre pour permettre une préservation locale du volume.*

On rappelle (voir partie 1.4.1) que dans le cas où S est représentée par un maillage triangulé, le volume du domaine défini par celui-ci peut s'exprimer analytiquement suivant l'eq. (1.23). Nous montrons ci-après que l'utilisation d'une formule spécialisée pour le cas de quadrangles peut se révéler nécessaire.

Calcul du volume pour un maillage quadrangulaire

Les maillages modélisés par les artistes par des approches paramétriques de type NURBS ou pour de l'animation sont généralement quadrangulaires [1]. Un exemple de tel maillage, que nous utiliserons dans nos applications, est présenté en fig. 2.8.

La triangulation triviale de ce type de maillage introduit un maillage de faible qualité [2] ainsi que des variations périodiques de l'aire du 1-voisinage des sommets.

Or, on peut noter que l'amplitude du gradient du volume dépend de l'aire du 1-voisinage des sommets.
En effet, il est facile de constater que dans le cas d'un maillage plan suivant (x, y), le gradient du volume est orienté suivant la normale z et sa valeur est égale à $A^v/3$, avec A^v l'aire du 1-voisinage du sommet.

Si cette aire varie périodiquement sur le maillage, cela engendre une oscillation non souhaitée de l'amplitude du gradient du volume. Ces oscillations d'amplitudes se répercutant en oscillations géométriques lors de l'application d'une correction du volume dépendant du calcul de cette amplitude. Un exemple de telles variations est donné en fig. 2.9.

Afin de résoudre le problème, nous proposons d'intégrer dans le calcul même du volume la gestion adéquate des quadrangles. Pour cela, nous considérons un quadrangle comme un morceau de patch bilinéaire (voir fig. 2.10).

1. Plus précisément, il s'agit de maillages mixtes triangles-quadrangles. Les quadrangles sont majoritaires, les triangles étant placés aux jonctions entre régions, de préférence dans les endroits peu mobiles ni visibles.
2. Les triangles issus de la triangulation d'un maillage quadrangulaire sont rectangles et non equilatéraux.

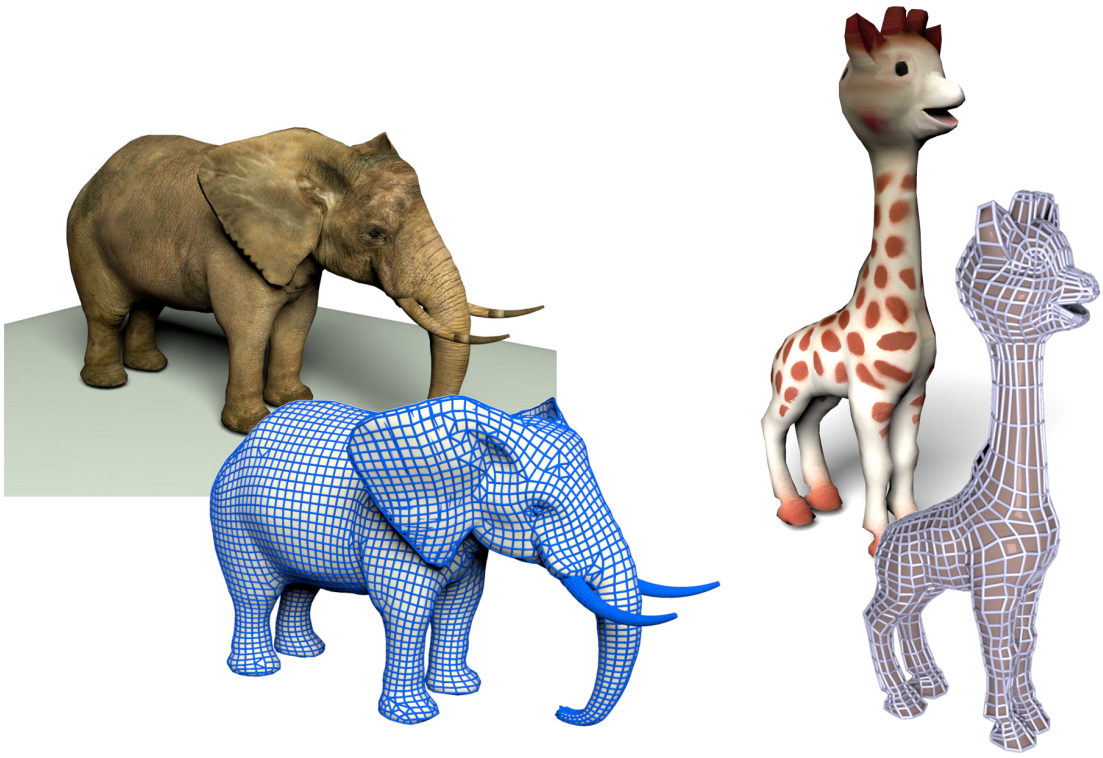


FIGURE 2.8 – *Exemples de maillages à dominante quadrangulaire utilisés dans nos résultats suivants.*

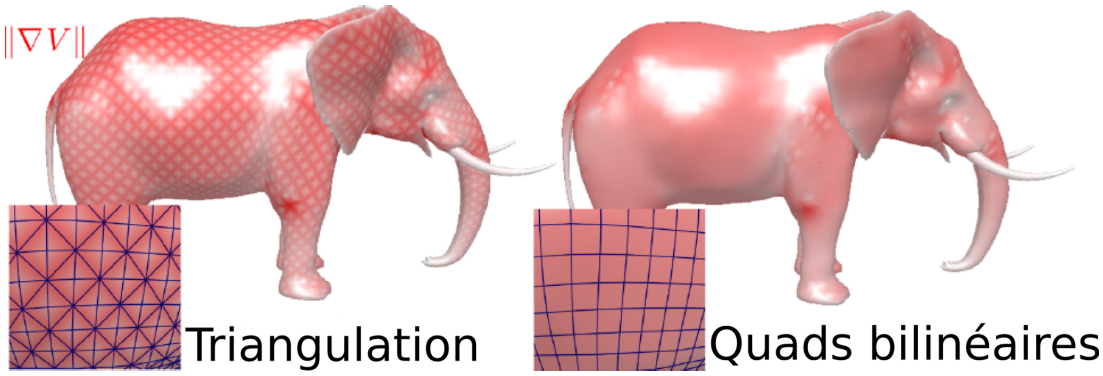


FIGURE 2.9 – *Impact de la triangulation sur la norme du gradient du volume* $\|\nabla V\|$*. À gauche, un maillage initialement quadrangulaire est triangulé. La variation périodique d'aire du 1-voisinage des sommets entraine la variation de la norme du gradient. À droite, le gradient est directement traité en prenant compte les quadrangles comme des morceaux de surfaces bilinéaires. La norme de celui-ci est alors plus lisse.*

En reconsidérant l'équation du volume pour une surface continue, il est possible d'exprimer celui-ci analytiquement pour ce type de surface. Supposons N_{quad} représenter le nombre de quadrangles du maillage, et un quadrangle d'indice i être formé par les 4 sommets

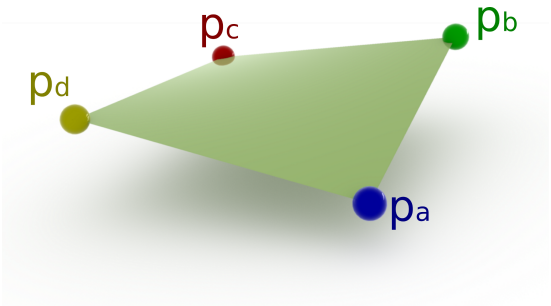


FIGURE 2.10 – *Exemple d'un quadrangle formé par 4 sommets* $(\mathbf{p}_a, \mathbf{p}_b, \mathbf{p}_c, \mathbf{p}_d)$ *considéré comme une surface bilinéaire.*

$(\mathbf{p}_{a_i}, \mathbf{p}_{b_i}, \mathbf{p}_{c_i}, \mathbf{p}_{d_i})$. Comme démontré en annexe A.4, l'expression du volume peut se mettre sous la forme

$$V = \sum_{i=1}^{N_{\text{quad}}} (\mathbf{q}_i^z)^T \, \mathrm{M} \, \mathbf{q}_i^d \,, \tag{2.1}$$

avec $\mathbf{q}_i^z = (z_{a_i}, z_{b_i}, z_{c_i}, z_{d_i})$,

$$\mathrm{M} = \frac{1}{36} \begin{pmatrix} 4 & 2 & 2 & 1 \\ 2 & 4 & 1 & 2 \\ 2 & 1 & 4 & 2 \\ 1 & 2 & 2 & 4 \end{pmatrix} ,$$

et enfin $\mathbf{q}_i^d = (q_{1_i}^d, q_{2_i}^d, q_{3_i}^d, q_{4_i}^d)$ le vecteur des déterminants

$$q_{1_i}^d = \begin{vmatrix} x_{b_i} - x_{a_i} & x_{d_i} - x_{a_i} \\ y_{b_i} - y_{a_i} & y_{d_i} - y_{a_i} \end{vmatrix} \quad q_{2_i}^d = \begin{vmatrix} x_{b_i} - x_{a_i} & x_{c_i} - x_{b_i} \\ y_{b_i} - y_{a_i} & y_{c_i} - y_{b_i} \end{vmatrix}$$

$$q_{3_i}^d = \begin{vmatrix} x_{c_i} - x_{d_i} & x_{d_i} - x_{a_i} \\ y_{c_i} - y_{d_i} & y_{d_i} - y_{a_i} \end{vmatrix} \quad q_{4_i}^d = \begin{vmatrix} x_{c_i} - x_{d_i} & x_{c_i} - x_{d_i} \\ y_{c_i} - y_{d_i} & y_{c_i} - y_{d_i} \end{vmatrix}$$

Le nouveau gradient de volume peut alors s'obtenir par dérivation analytique de cette expression.

Enfin, pour un maillage mixte formé à la fois de N_{tri} triangles et N_{quad} quadrangles, le volume total V peut s'exprimer comme somme des sous-volumes issus des triangles, et ceux issus des quadrangles.

Discussion de l'approche globale

On note que le calcul du changement de volume global peut aisément s'exprimer analytiquement sur des maillages quelconques. Cependant, cette relation ne permet pas l'estimation directe de la localisation du changement de volume. La même perte de volume au niveau du bras d'un personnage et d'un pied donne en effet le même résultat sur ΔV.

Nous proposons deux approches différentes permettant d'estimer ce changement de volume de manière locale dans le cas d'une déformation par squelette.

1. L'estimation approximée et rapide de ce changement de volume sur des régions segmentées du maillage
2. Le calcul exact de la variation de volume en N_j étapes (avec N_j le nombre de joints du squelette d'animation), mais plus couteux.

2.2.1 Estimation approximée rapide d'un changement de volume par découpage en prismes

Méthode

Supposons la surface $\overline{S}$ être segmentée en parties distinctes. Une segmentation triviale convenant à notre cas consiste à définir une région R_i comme l'ensemble des positions dont le poids de skinning maximal est lié à un os i donné (voir fig. 2.11 gauche). Le principe de l'estimation de ΔV locale consiste alors à estimer le changement de volume sur chacune de ces régions.

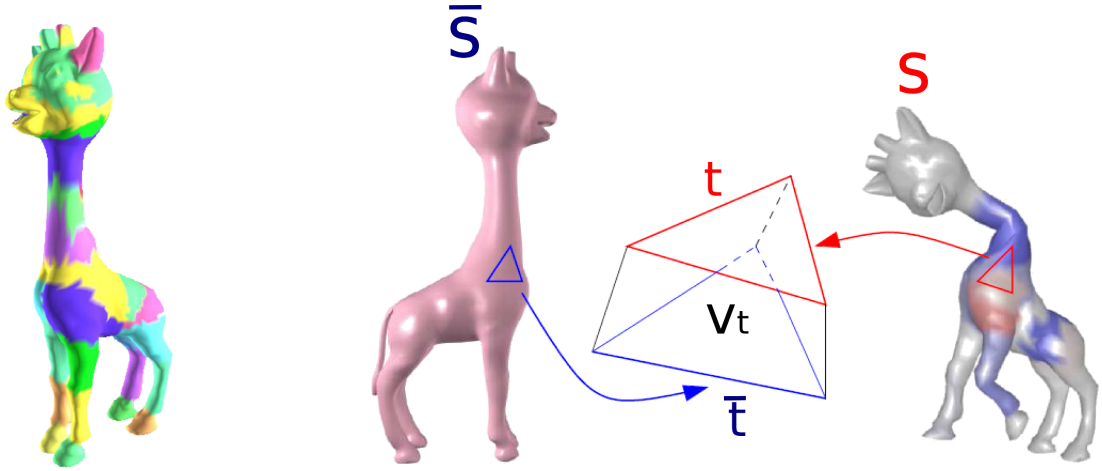


FIGURE 2.11 – *Gauche : Segmentation possible d'une surface S, chaque couleur indique une région R_i. Droite : Estimation du changement de volume local par somme de sous-volumes de prismes. Un triangle de $\overline{t} \in \overline{S}$ et un triangle $t \in S$ sont comparés dans la base locale associée à leur région respective. Ces deux triangles forment un prisme de volume v_t. Le changement de volume de la région est obtenu en sommant l'ensemble de ces sous volumes. Un gain de volume local est indiqué en rouge, alors qu'une perte de volume est indiquée en bleu sur la figure de droite.*

Étant donné que ces régions géométriques possèdent des bords, elles ne définissent plus de domaine fermé de l'espace, et donc n'ont pas de volume exact. L'équation (2.11) n'est ainsi plus valide. Nous proposons à la place une estimation du changement de volume ΔV_{R_i} dû au gonflement/dégonflement local de la surface sur la région R_i.

Supposons $S_{|R_i}$ (resp. $\overline{S}_{|R_i}$) la surface déformée (resp. originale) restreinte à la région liée à l'os i. En notant T_i la déformation de cet os, les surfaces exprimées dans le repère local de l'os i sont donnés par $S^{Loc}_{|R_i} = \mathrm{T}_i^{-1}\, S_{|R_i}$ (resp. $\overline{S}^{Loc}_{|R_i} = \overline{\mathrm{T}}_i^{-1}\, \overline{S}_{|R_i}$).

Pour calculer la variation de volume associée à ces deux morceaux de surface, construisons le prisme dont les deux bases triangulaires sont formées du triangle $\overline{t}$ de $\overline{S}^{Loc}_{|R_i}$ et du triangle t

de $S^{Loc}_{|R_i}$. Enfin, considérons son volume signé Δv_t (voir fig. 2.11). Le changement de volume local associé à la région R_i est alors donné par la somme de ces sous-volumes :

$$V_{R_i} = \sum_{\text{triangle } t \in S_{|R_i}} v_t \ .$$

Notons que les faces quadrangulaires du prisme peuvent être triangulées en considérant l'ajout d'un sommet au barycentre de ces faces pour l'ensemble des prismes comme illustré en fig. 2.12.

On notera également que cette expression revient à former et calculer le volume exact de la surface fermée délimitée par $S^{Loc}_{|R_i}$ et $\overline{S}^{Loc}_{|R_i}$, et dont les bords ont été fermés par des quadrangles joignant les bords des triangles respectifs des deux surfaces.

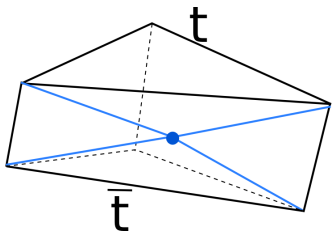


FIGURE 2.12 – *Triangulation homogène des faces des prismes par l'ajout d'un sommet central.*

Discussion

Une fois le calcul effectué sur chacune des régions de la segmentation, on obtient une estimation du changement signé de volume local. Il s'agit uniquement d'une approximation et dans le cas général : $\sum_i V_{R_i} \neq V_S - V_{\overline{S}}$. Notons cependant que l'approximation permet de transcrire avec une bonne précision les gonflements et dégonflements locaux (déplacement des triangles dans la direction orthogonale à l'axe de l'os). Au contraire, les glissements de surface dans ce même repère ne sont pas interprétés correctement.
Un résultat est montré en fig. 2.11 droite, où le calcul local de la perte/gain du volume est encodé en couleur. Pour des rotations d'angles variants entre 0 et 90 degrés, l'approximation de cette méthode appliquée sur le coup de la girafe entraine une erreur de calcul de volume de l'ordre de 20%.

Cette méthode est donc adaptée dans le cas classique du skinning comportant de manière quasi exclusive des rotations des joints générant des creusements de la surface. Si l'on souhaite par contre pouvoir modéliser le cas d'allongements d'os (extension d'un bras en caoutchouc, etc), une autre approche se révèle nécessaire.

En terme de cout de calcul, l'estimation de la variation de volume sur chacune des parties est du même ordre que le calcul du changement de volume global. Il consiste en effet en une unique passe de calcul sur chaque triangle de la surface et sa complexité est donc en $\mathcal{O}(N_t)$, avec N_t le nombre de triangles.

Nous avons présenté une méthode d'estimation approchée de la variation de volume localisée autour de chaque os. L'algorithme est aussi rapide que le calcul standard de la variation de volume globale, mais n'est pas exact pour des déformations quelconques. Nous proposons une seconde méthode de calcul du volume local moins rapide, mais cette fois exacte pour tout type de déformation à squelette.

2.2.2 Calcul exact d'un changement de volume par itération sur le squelette d'animation

La méthode précédente proposait une segmentation spatiale du maillage déformé de manière globale. À l'inverse, dans le second cas, nous allons considérer la géométrie du maillage globalement, mais déformer localement celle-ci.

Méthode

Soit la suite de surfaces $(S_k)_{k\in[\![1,N_j]\!]}$, avec N_j le nombre de joints du squelette d'animation. Pour k fixé, S_k est définie comme la surface déformée par skinning en considérant uniquement la déformation des joints $[\![1,k]\!]$. On a alors $S_0 = \overline{S}$ et $S_{N_j} = S$.

Notons que l'ordre des joints peut être arbitraire à condition que la suite des joints $(k)_{k\in[\![1,N_j]\!]}$ suive un parcours hiérarchique valide du squelette d'animation.

En calculant globalement le changement de volume $\Delta V_k = V_{S_k} - V_{S_{k-1}}$, on obtient la variation du volume engendrée par la déformation du joint k. Il s'agit donc du changement de volume local lié à un joint donné comme illustré en fig. 2.13. De plus, ce calcul de volume est par définition exact, car

$$\sum_{k=1}^{N_j} \Delta V_k = V_{S_{N_j}} - V_{S_0} = V_S - V_{\overline{S}}\,.$$

Discussion

Cette méthode de calcul de changement local de volume possède trois avantages par rapport à l'estimation approximée présentée précédemment

- **Précision :** Le calcul est exact au sens où les changements de volume locaux se somment exactement au changement du volume global engendré par le skinning final.
- **Robustesse :** Aucune région arbitraire de segmentation n'a besoin d'être définie, ce qui autorise les déformations de chaque joint à posséder un rayon d'influence non borné.
- **Adaptabilité :** L'approche étant robuste, elle peut s'adapter à tout type de déformation par squelette.

On notera cependant que cette approche est plus lourde en temps de calcul, en effet l'algorithme repose sur N_j calculs itératifs de volume global en plus de nécessiter le calcul explicite des N_j déformations successives par skinning. La complexité algorithmique associée est donc du type $\mathcal{O}(N_j(N_t + N_s))$, avec N_s nombre de sommets du maillage (issu de la déformation par skinning intermédiaire).

De plus, le résultat du calcul dépend de l'ordre suivi pour parcourir le squelette. En effet, lorsque les joints possèdent de grands rayons d'influences, les déformations liées à un joint peuvent se superposer à ceux d'un autre. La part de déformation associée à un joint peut ainsi varier, et nous n'avons donc pas unicité de notre définition de changement de volume local. Nous pouvons ainsi définir N_{parcours} suites $(\Delta V_k)_{k\in[\![1,N_j]\!]}$ possibles de valeurs de changement de volume local exact admissible, où N_{parcours} correspond au nombre de parcours hiérarchiques possible du graphe du squelette. Nous verrons cependant lors des résultats, que l'influence de ce chemin suivit reste faible et les différents résultats obtenus restent visuellement plausibles.

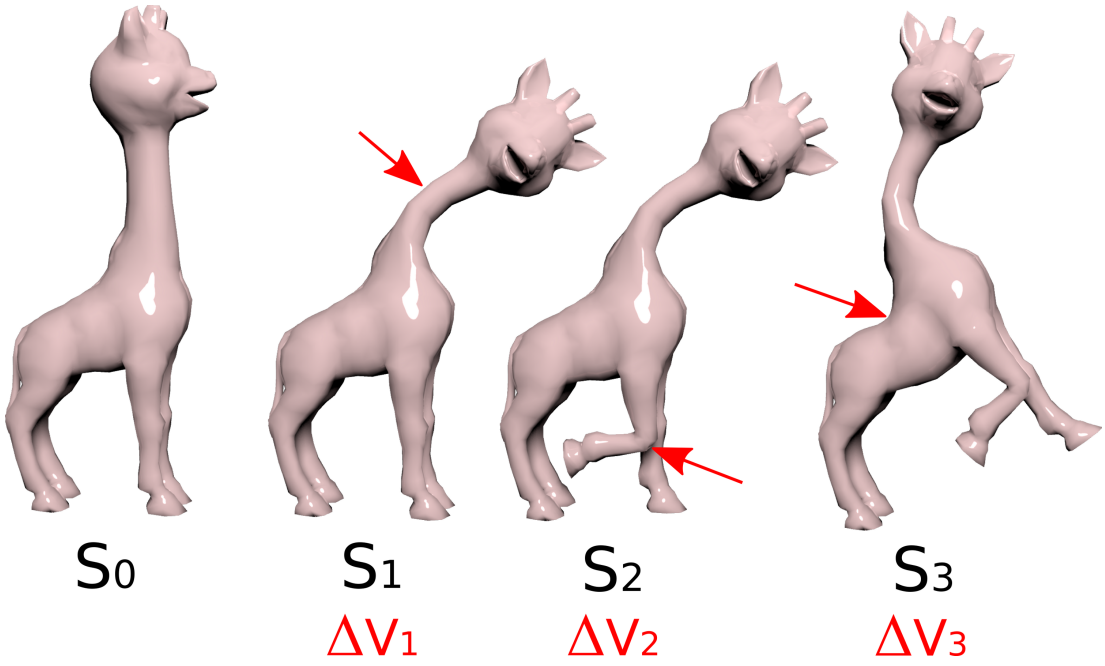


FIGURE 2.13 – *Illustration du calcul itératif exact du changement de volume pour un skinning réalisé avec 3 rotations. La surface originale est S_0, alors que S_3 est la surface skinné finale. En considérant les changements de volume (calculés globalement) pour les étapes intermédiaires S_1 et S_2, puis enfin S_3, on peut connaitre la part exacte de volume modifiée par chaque articulation indépendamment. On accède ainsi à la connaissance du changement local du volume ΔV_i pour chaque joint i.*

2.3 PRESERVATION LOCALE DE VOLUME

Nous avons présenté deux approches permettant d'estimer de manière exacte ou approchée la variation de volume locale d'un objet déformé par une approche de type *squelette d'animation*.

Nous proposons d'utiliser l'information de changement local de volume pour corriger celui-ci lors de la déformation. Cela permet de définir une déformation par skinning à volume constant.

Nous généraliserons directement l'approche dans le cas où l'on souhaite fixer le volume à une valeur définie dénommée V_{target} et non simplement préserver le volume initial. Des modélisations expressives pourront ainsi exagérer le gonflement d'un muscle par exemple. Pour une déformation à volume constant, il suffira d'utiliser $V_{\text{target}} = V_{\overline{S}}$.

Nous présentons, comme précédemment, deux approches possibles de correction en volume suivant l'application visée. Nous détaillons une première méthode permettant de fixer le volume final de manière rapide, mais uniquement approchée. Puis, nous présentons une seconde approche en trois passes locales permettant d'atteindre le volume fixé de manière exacte.

2.3.1 Préservation de volume approximée en une passe globale

On rappelle que la méthode de préservation de volume introduite par Sauvage *et al.* [SHB07] présentée en partie 1.4.2 permet la préservation de volume global approché d'un maillage déformé. L'approche ne permet cependant pas de traiter adéquatement le cas de modification de volume locale comme celle présentée en fig. 2.7.

Nous nous proposons d'étendre l'approche de Sauvage *et al.* pour rendre la correction locale en modifiant la norme utilisée.

En s'inspirant de l'eq. (1.26), nous cherchons cette fois dans un premier temps à déformer la surface dans la direction normale (voir fig. 2.14) telle que la contrainte de volume soit vérifiée. En réutilisant les notations de l'équation originale, on a alors à résoudre

$$\begin{cases} \min & \|\boldsymbol{\rho}\|^2_{\mathbb{R}^N} \\ \text{contraint à} & V(\mathbf{P}+\boldsymbol{\rho}\mathbf{N}) = V_{\text{target}}\ , \end{cases} \tag{2.2}$$

avec $\rho = (\rho_1, \cdots, \rho_{N_s}) \in \mathbb{R}^{N_s}$ le vecteur d'amplitude de déformation à appliquer le long de la normale pour chaque position. La norme $\|\ \|_{\mathbb{R}^{N_s}}$ étant celle issue du produit scalaire canonique de $\mathbb{R}^{N_s}$. Minimiser sa norme revient donc à déformer la surface le moins possible. La notation $\boldsymbol{\rho}\mathbf{N}$ signifie le vecteur résultant du produit correspondant $\boldsymbol{\rho}\mathbf{N} = (\rho_1\mathbf{n}_1, \cdots, \rho_{N_s}\mathbf{n}_{N_s})$.

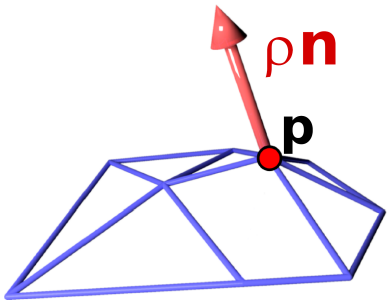


FIGURE 2.14 – *Déformation de la surface suivant le vecteur normal* $\mathbf{n}$ *d'amplitude* ρ*. L'amplitude* ρ *doit être calculée tel que le volume de la surface déformée soit égale à* V_{target}*.*

L'approche naïve consiste à traiter l'ensemble des sommets de la surface de manière identique ce qui résulte en une préservation de volume globale. Celle-ci n'est pas satisfaisante comme le montre la fig. 2.15 où une telle préservation globale est appliquée sur un cas d'école (la résolution d'un tel système est présentée dans la suite du document).

Afin d'améliorer le résultat visuel, on peut ajouter un degré de contrôle supplémentaire permettant de forcer la correction à agir localement aux endroits qui ont initialement perdu du volume, ou même de manière artistiquement guidé. Pour cela, nous allons pondérer le déplacement des sommets en fonction de leur position sur la surface. Nous introduisons pour cela un champ scalaire continu de pondération $\gamma > 0$ défini sur la surface telle qu'une valeur importante de γ va imposer une contrainte de position forte (partie rigide de la surface), alors qu'une valeur faible de γ indiquera que ce sommet sera faiblement contraint (partie déformable de la surface).

Dans le cas discret du maillage, nous introduisons ainsi un vecteur de poids de pondérations par sommets : $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_{N_s}) \in (\mathbb{R}^{\star+})^{N_s}$ représentatif du champ scalaire défini sur la surface. Afin de tenir compte de la contrainte imposée par cette pondération, nous modifions la

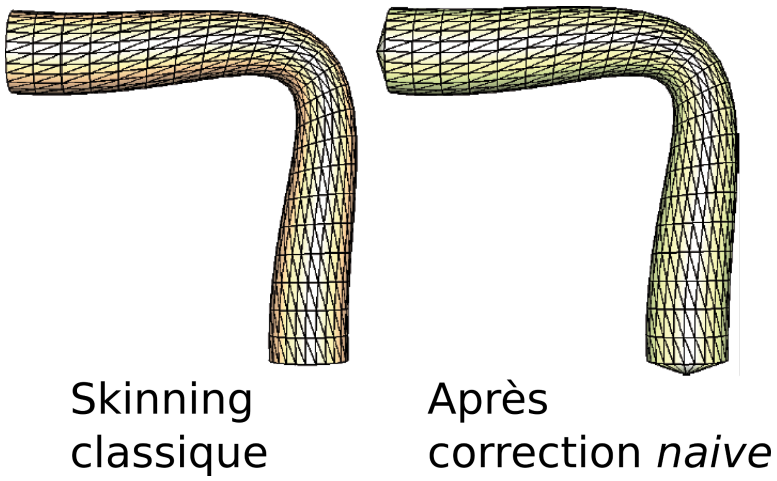


FIGURE 2.15 – *Approche naïve d'une application de préservation de volume suivant l'eq. (2.2) lorsque les sommets sont traités uniformément. Le cylindre de gauche déformé par skinning montre clairement une contraction de son rayon au niveau de l'articulation. La surface de droite, après correction, possède globalement le volume du cylindre non plié, par contre l'articulation centrale possède toujours un rayon trop faible, le volume n'est pas préservé localement. De plus, les sommets extrémaux du cylindre ont été largement déplacés ce qui n'est généralement pas l'effet escompté.*

norme utilisée en considérant désormais

$$\|\rho\|_{\boldsymbol{\gamma}} = \sum_{k=1}^{N_s} \frac{\rho_k^2}{\gamma_k} \, .$$

La nouvelle équation de minimisation est alors exprimable par :

$$\left\{ \begin{array}{ll} \min & \|\boldsymbol{\rho}\|_{\gamma}^2 \\ \text{contraint à} & V(\mathbf{P} + \boldsymbol{\rho}\mathbf{N}) = V_{\text{target}} \, . \end{array} \right. \tag{2.3}$$

Similairement à l'approche expliquée en partie 1.4.2, nous linéarisons la contrainte de volume suivant suivant la variable ρ. L'approche de résolution du système sous contrainte de Sauvage *et al.* [SHB07] peut alors être appliquée pour exprimer la solution analytique (la démonstration complète est disponible en annexe B.2).

$$\forall k \in [\![1, N_s]\!] \, , \quad \rho_k = \Delta V \frac{\gamma_k < \mathbf{n}_k, \nabla V(\mathbf{p}_k) >}{\sum_{j=1}^{N} \gamma_j < \mathbf{n}_j, \nabla V(\mathbf{p}_j) >^2} \, . \tag{2.4}$$

Combiné avec l'estimation locale de la variation de volume ΔV, l'équation 2.4 permet de corriger la quantité de volume souhaité vis-à-vis d'un os ou d'une région donnée, tout en imposant une distribution particulière de cette correction grâce à la pondération par les poids γ.

Notons que d'un point de vue algorithmique, la correction pondérée possède la même complexité que le cas de la correction originale globale. L'étape la plus couteuse étant le calcul du gradient du volume pour chaque sommet, la complexité totale reste linéaire en fonction du nombre de sommets du maillage.

2.3.2 Préservation de volume exacte multi-passes suivant des axes locaux

Méthode de correction en trois étapes

La correction précédente est basée sur la linéarisation de la contrainte de volume. La valeur numérique du volume souhaité n'est donc, dans le cas général, pas atteinte exactement. Afin d'assurer une préservation exacte du volume, nous étendons l'approche de Sauvage *et al.* [SHB07]. Cependant, dans notre approche, chaque axe est traité de manière séquentielle, et non en parallèle. Cette correction se déroule en trois étapes successives et à été initialement utilisée par Elber [Elb00] pour la préservation de contraintes bi et tri-linéaires appliquée sur des courbes et surfaces de Bézier. Nous étendons ici sont application dans le cas de maillages triangulés.

Notre approche est la suivante (voir fig. 2.16) :

1. La surface S est tout d'abord déformée en $S^\star$ suivant l'axe x afin de corriger un pourcentage $\mu_x \Delta V$ de l'erreur en volume.
2. Cette forme intermédiaire $S^\star$ sert de point de départ pour l'étape de déformation suivant l'axe y aboutissant à $S^{\star\star}$ afin de corriger un pourcentage $\mu_y \Delta V$ de l'erreur en volume.
3. Cette dernière surface $S^{\star\star}$ est finalement déformée suivant l'axe z pour aboutir à S_1, corrigeant alors un pourcentage $\mu_z \Delta V$ de l'erreur en volume.

Si $\mu_x + \mu_y + \mu_z = 1$, on a alors exactement $V(S_1) = V(S) + \Delta V = V_{\text{target}}$ ce qui permet la mise en place d'un contrôle ou d'une préservation exacte du volume souhaité.

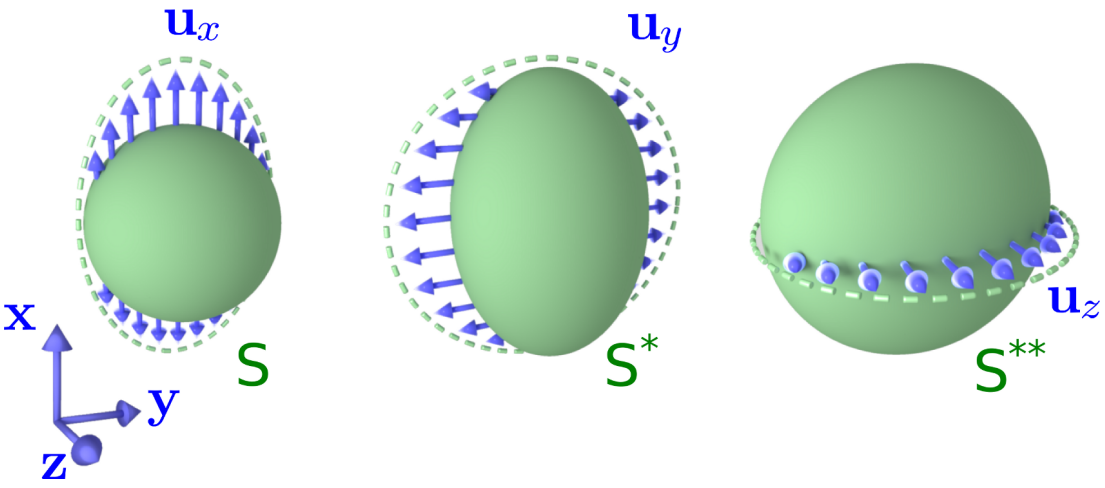


FIGURE 2.16 – *Correction de volume exacte en trois étapes successives suivant x, y puis z pour un objet d'exemple sphérique. Le résultat de l'étape précédente servant de point de départ à l'étape courante.*

En utilisant la pondération γ introduite précédemment, la minimisation que nous appliquons s'exprime par

$$\begin{cases} \min & \|\mathbf{U}\|_\gamma^2 \\ \text{contraint à} & V(\mathbf{P}+\mathbf{U}) = V_{\text{target}} \, . \end{cases} \tag{2.5}$$

En suivant cette approche en trois étapes, la solution exacte analytique, dont la démonstration est présentée en annexe B.3, peut alors s'exprimer par :

$$\forall k \in [\![1, N_s]\!] \ , \quad \mathbf{u}_k = \gamma_k \, \Delta V \left(\mu_x \frac{\nabla_{\mathbf{x}} V(\mathbf{p}_k)}{\|\nabla_{\mathbf{x}} V\|^2_{1/\gamma}}, \mu_y \frac{\nabla_{\mathbf{y}} V^\star(\mathbf{p}_k)}{\|\nabla_{\mathbf{y}} V^\star\|^2_{1/\gamma}}, \mu_z \frac{\nabla_{\mathbf{z}} V^{\star\star}(\mathbf{p}_k)}{\|\nabla_{\mathbf{z}} V^{\star\star}\|^2_{1/\gamma}} \right) \ , \quad (2.6)$$

avec la norme de pondération locale $\|\mathbf{a}\|^2_{1/\gamma} = \sum_{k=1}^{N_s} \gamma_k a_k^2$. $V^\star$ et $V^{\star\star}$ désignent respectivement les volumes calculés à partir des surfaces intermédiaires obtenues après la première et la seconde étape de déformation.

Application d'axes locaux

Notons que l'utilisation d'axes globaux fixes de déformation (x, y, z) n'est cependant pas adapté au cas d'un objet articulé ayant différentes orientations locales. Pour remédier à ce défaut, nous proposons d'appliquer ces étapes de corrections locales le long d'axes de corrections choisis de manière appropriée vis-à-vis de l'orientation locale de la surface.

Soit (x^L, y^L, z^L) les coordonnées exprimées dans un repère local. La solution donnée en eq. (2.6) peut alors s'appliquer dans ce système d'axe. Afin de prendre avantage de cette localité, il est possible d'appliquer cette correction dans différentes régions de l'objet avec, pour chaque région, un système d'axe local propre.

Lors de l'animation d'un personnage articulé, la correction locale exacte se réalisera ainsi séquentiellement pour chaque os du squelette d'animation, avec, pour chaque articulation, un changement de volume calculé exactement suivant l'approche présentée précédemment.

Discussion

Cette étape de correction est exacte au sens où le volume final de la surface est V_{target} indépendamment du choix des paramètres.

On notera que similairement à l'approche itérative exacte du calcul de changement de volume local (partie 2.2.2), cette correction est également dépendante de l'ordre de parcours suivant x, y et z. Nous montrerons également lors de l'analyse des résultats que cet impact reste visuellement limité.

Enfin, cette approche permet la mise en place de trois degrés de liberté au niveau des paramètres de contrôle afin de modifier l'apparence finale tout en vérifiant la contrainte de volume :

1. **La pondération de la norme** γ. Ce paramètre permet de définir précisément la *localisation du gonflement/dégonflement* introduit par la correction de volume. Nous montrons dans la section suivante comment définir ces valeurs de manière intuitive.
2. **Les ratios de correction** (μ_x, μ_y, μ_z) permettent de manipuler le *pourcentage d'anisotropie* de la déformation en privilégiant certains axes plutôt que d'autres. Par exemple, le gonflement d'un muscle doit être modélisé par une déformation anisotrope.
3. **Le choix des axes locaux** (x^L, y^L, z^L) permet de définir les *directions des axes de déformation* de manière appropriée dans le cas de corrections anisotropes. Il est commun d'avoir des axes locaux différents pour chaque partie de l'objet. Dans ce cas, la correction de volume sera appliquée différemment pour chaque partie en adaptant à chaque fois ce repère. La section suivante explique également comment définir de manière automatique ces directions de déformations locales.

Nous avons présenté deux méthodes de correction de position des sommets d'un maillage permettant de garantir que la surface associée délimite un volume V_{target} fixé localement. Ces deux méthodes permettent de gérer la contrainte de manière soit approchée, soit exacte. Nous avons pour cela introduit différents paramètres permettant de localiser et d'orienter cette correction. Dans la partie suivante, nous expliquons comment définir ces paramètres de contrôle local.

2.4 CONTROLE DE L'APPARENCE FINALE

Dans un premier temps, nous proposons une méthode de définition entièrement automatique des poids de pondération γ dans le cas de formes organiques en nous basant sur des a-priori anatomiques. Dans un second temps, nous proposons une approche alternative permettant des corrections anisotropes contrôlables intuitivement par l'utilisateur suivant des courbes de profil.

2.4.1 Gestion automatique par poids de skinning et distance au squelette

On s'intéresse premièrement au cas particulier de formes organiques telles qu'un doigt plié, ou le cou d'un animal. Nous proposons pour cela de considérer le cas d'une correction de volume isotrope pour laquelle la distribution de déformation sera automatiquement définie. L'objectif est de construire une méthode de déformation par skinning à volume constant à partir de la seule donnée d'entrée d'un maillage d'un personnage ou d'un animal.

Localisation

Les poids de pondération γ définissent la distribution spatiale de la correction en volume. Il faut fixer des valeurs de γ faible aux endroits où l'on souhaite une correction de position importante, et inversement dans le cas où l'on souhaite garder les positions de la surface inchangées.

Nous partons du constat que les pertes de volume engendrées par la déformation par skinning sont localisées aux articulations des membres. En effet, lorsqu'au moins deux os se rencontrent, les influences de plusieurs repères rigides entrent en compte. Il en résulte une l'interpolation linéaire des matrices de transformation qui leur sont associées. Ce sont donc ces zones qui subissent les artefacts du skinning, les zones liées à un unique joint étant rigidement déformées. La figure 2.17 présente une illustration de ces zones pour un cylindre skinné.

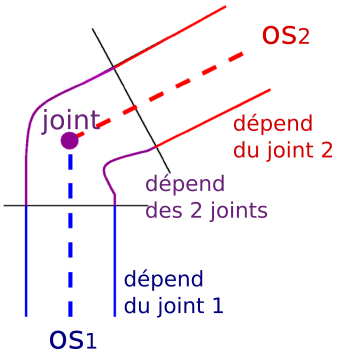


FIGURE 2.17 – *Définition automatique des régions nécessitant une compensation de volume. Les régions subissant des pertes de volumes liées à la méthode du skinning sont celles situées autour des articulations, car elles sont liées à plusieurs os.*

Considérons le poids de skinning maximal d'un sommet k donné par rapport à l'ensemble des os dont il dépend. Les sommets appartenant à ces zones fortement déformées sont caractérisés par un poids de skinning maximal $\omega_k^{\max} := \max_{\text{joints } j} \omega_{kj}$ proche ou inférieur à 0.5. Au contraire des zones déformées rigidement pour lesquels $\omega_k^{\max}$ est proche de 1.

Une première définition automatique des poids de pondération γ consiste ainsi à les relier directement au poids de skinning maximal avec $\gamma_k = (1 - \omega_k^{\max})^a$ où a un paramètre de contrôle (généralement 1 ou 2).

A-priori anatomiques

Cette distribution de poids automatique permet de localiser l'articulation, mais ne prend cependant pas en compte de paramètres anatomiques tels que le placement non relatif des os dans la forme. Par exemple, les os du coude ou d'un doigt (voir fig. 2.18) sont placés de manière très excentrée de par la distribution non radiale de muscle ou de chair.

La propension aux parties de chair *molle* à se déformer plus aisément n'est pas forcément traduite par la donnée des poids de skinning. L'utilisation de la définition précédente des poids de pondération γ engendre ainsi une correction ayant une distribution radiale autour des os. Pour tenir compte de cet a-priori anatomique, nous proposons d'introduire la distance au squelette comme paramètre supplémentaire de la pondération. Une distance au squelette faible modélise une faible quantité de chair, et donc une contrainte de plus importante sur la position des sommets.

Soit $d_k^{\min}$ la distance minimale entre le sommet k et l'os d'influence prépondérante (c'est à dire, tenant compte des informations de topologie comme présentée en partie 2.1). Nous modélisons la pondération finale par

$$\gamma_k = (1 - \omega_k^{\max})^a \left(d_k^{\min}\right)^b . \tag{2.7}$$

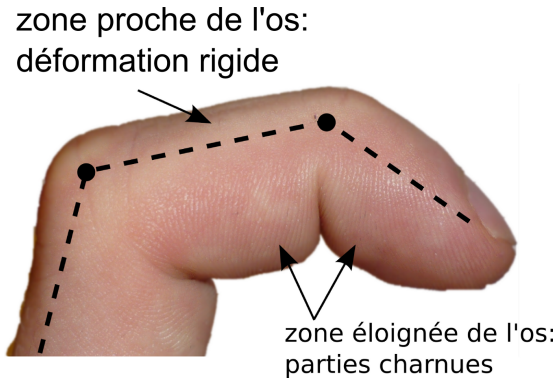


FIGURE 2.18 – *Exemple de déformations réelle dépendante de la position excentrée d'un os. La partie dorsale du doigt où la peau est proche de l'os se déforme rigidement avec celui-ci. La partie ventrale, éloignée de l'os, gonfle et se déforme non rigidement.*

Un exemple de résultat obtenu sur un cylindre est illustré en fig. 2.19.

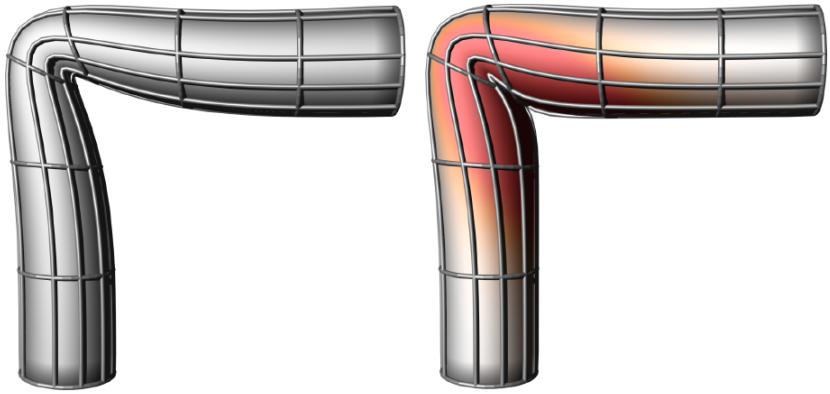


FIGURE 2.19 – *Résultat obtenu de préservation de volume (méthode issue de l'eq. (2.4)) après pondération automatique par les poids γ automatiques. Le cylindre possède un squelette d'animation excentré permettant de donner ce gonflement unilatéral de type organique. L'encodage couleur indique la distribution des poids γ sur la surface.*

Notons que cette méthode est d'autant plus efficace si le squelette d'animation est proche du squelette anatomique avec un positionnement excentré des os.

2.4.2 Controle utilisateur par courbes de profils

L'approche précédente permet de modéliser des formes organiques simples de manière automatique si le squelette d'animation est proche d'un squelette anatomique. On peut cependant chercher à définir des déformations plus complexes en possédant un contrôle précis de la forme de la correction que l'on cherche à donner. Nous pouvons par exemple chercher à modéliser le comportement d'un muscle du biceps ayant lieu sur un os donné de manière non symétrique. Nous pouvons également chercher à complexifier le profil de la correction en introduisant des bourrelets dans le cas du ventre d'un animal s'accroupissant.

Pour modéliser ces variations importantes de paramètres, nous considérons qu'il est nécessaire de laisser suffisamment de liberté à un artiste pour déterminer lui même la forme finale souhaitée de manière intuitive. Inspiré par la démarche de Nealen *et al.* [ANCO05] nous proposons de définir la forme de la distribution géométrique de la correction de volume à l'aide de courbes de profils dans des directions privilégiées. La déformation des sommets vient alors suivre la forme de cette distribution, alors que l'intensité du déplacement est quant à elle automatiquement définie par la contrainte de volume.

Nous établissons un compromis entre contrôle et automatisation. Pour cela, nous choisissons de définir :

- des axes (x^L, y^L, z^L) de direction de déformation locales appropriés de manière automatique d'après le squelette d'animation.
- un choix manuel des proportions du taux d'anisotropie suivant chacun des trois axes (μ_x, μ_y, μ_z).
- une définition semi-automatique de la distribution spatiale de la correction de volume à l'aide des courbes de silhouette définies par l'utilisateur.

Définition des axes de directions

Nous cherchons à définir un repère local associé à chaque membre (os) d'un personnage virtuel s'adaptant automatiquement à l'animation afin de générer des gonflements ou dégonflements anisotropes suivant des axes appropriés.
Considérons le squelette d'animation hiérarchique associé au skinning du personnage virtuel. Pour chaque membre k du personnage, une direction principale triviale possible $\mathbf{u}_x^k$ est donnée par la direction de l'os parent au joint. Une seconde direction principale $\mathbf{u}_y^k$ est donnée par l'axe de rotation de ce joint lors de l'animation par skinning. Enfin, on définit la dernière direction principale pour former une base orthonormale avec $\mathbf{u}_z^k := \mathbf{u}_x^k \times \mathbf{u}_y^k$.

Dans le cas d'une surface tubulaire, cette base locale s'interprète géométriquement (voir fig. 2.20) : L'axe $\mathbf{u}_y$ permettant de définir une direction orthogonale à la rotation, il permet de définir l'orientation principale du gonflement typique d'un objet en caoutchouc. A l'inverse, l'axe $\mathbf{u}_z$ permet de définir l'axe privilégié de gonflement d'un muscle.

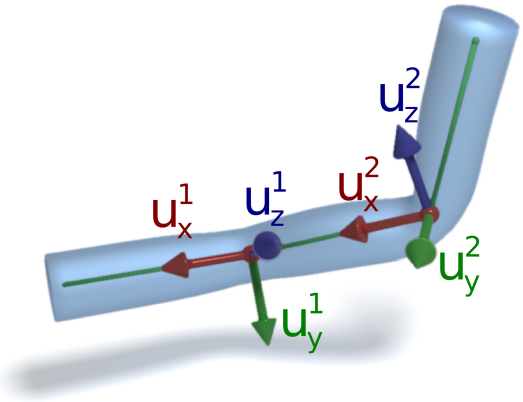


FIGURE 2.20 – *Repères locaux associés à un cylindre. Ces repères sont automatiquement définis et évoluent en fonction de l'articulation du squelette d'animation.*

Proportion d'anisotropie

Étant donnée la définition précédante des axes privilégiés de déformation, l'utilisateur est libre de définir l'importance relative suivant chacun de ces axes grâce aux paramètres (μ_x, μ_y, μ_z). Ainsi un gonflement isotrope autour d'un joint pourra se modéliser par $(\mu_x, \mu_y, \mu_z) = (1, 1, 1)/3$. Au contraire, un effet très anisotrope modélisant le plis d'un cylindre en caoutchouc comme celui illustré en fig. 2.21 pourra se modéliser directement par $(\mu_x, \mu_y, \mu_z) = (0, 1, 0)$.

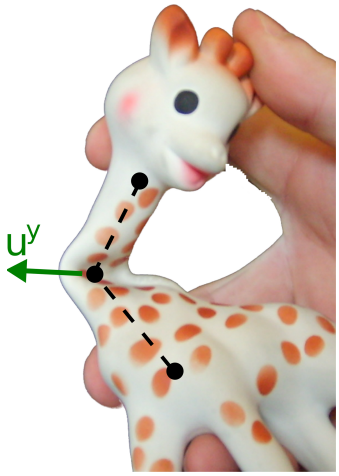


FIGURE 2.21 – *Exemple de déformation anisotrope. Le cou de ce jouet en caoutchouc gonfle de manière anisotrope lorsque celui-ci est plié. La direction de gonflement est donnée par l'axe de la rotation, c'est à dire* $\mathbf{u}_y$*. Une pondération de déformation du type* $(\mu_x, \mu_y, \mu_z) = (0, 1, 0)$ *permet de modéliser ce type d'effet.*

Distribution spatiale de correction à l'aide de courbes de profils

La distribution géométrique des poids γ associés à un os donné permet de définir finement la silhouette de la correction générée pour chaque partie locale de l'objet animé.

Par défaut une correction locale sera définie comme une pondération d'influence décroissante en fonction de la distance au joint racine de l'os donné. On peut donc définir pour ce joint, la pondération du sommet k suivant une Gaussienne

$$\gamma_k^{\text{defaut}}(\mathbf{p}_k^L, \sigma) := \exp\left(-\frac{\left\|\mathbf{p}_k^L\right\|^2}{\sigma^2}\right) , \tag{2.8}$$

où $\mathbf{p}_k^L$ sont les coordonnées relative du sommet k dans le repère local associé, et σ permet de paramétrer la vitesse de décroissance de cette pondération. Classiquement, nous considérons $\sigma = L_b/4$, où L_b représente la longueur de l'os. Un os de faible longueur sera ainsi associé à une correction plus localisée qu'une déformation induite par un os de grande taille.

Nous proposons par la suite de moduler cette distribution Gaussienne en considérant

$$\gamma_k(\mathbf{p}_k^L) = A_x(x_k^L)A_y(y_k^L)A_z(z_k^L)\ \gamma_k^{\text{defaut}}(\mathbf{p}_k^L - \mathbf{t}, \sigma) . \tag{2.9}$$

Ici A sont des fonctions unidimensionnelles représentant les courbes de silhouettes de la forme finale suivant les 3 axes orthogonaux. $\mathbf{t}$ est quant à lui un offset de translation permettant d'excenter la décroissance de la pondération (par exemple, un biceps pourra être modélisé par un gonflement dont le maximum est excentré au milieu de l'os).

Nous détaillons ci-après les paramètres choisis pour modéliser différents effets caractéristiques. Les différents résultats sont visibles en fig. 2.22.

Inflation isotrope locale

Considérons un cylindre pouvant se plier sans modification locale apparente de son volume, et donc de son rayon. Il s'agit d'un effet obtenu avec les paramètres par défaut de notre approche.

- Le gonflement est centré au niveau du joint générant la rotation $\mathbf{t} = 0$.
- Le gonflement à lieu de manière isotrope $(\mu_x, \mu_y, \mu_z) = (1, 1, 1)/3$.
- La distribution du profil de pondération suit la décroissance Gaussienne par défaut : $\gamma = \gamma^{\text{defaut}}$.

Un tel effet est présenté en fig. 2.22 b.

Inflation anisotrope locale

Considérons un objet tubulaire creux en caoutchouc. Lorsqu'un tel objet est plié, il génère une forme identifiable par une extension locale au niveau du coude dans la direction de l'axe de rotation du plis. Cet effet peut se modéliser par un skinning à volume constant tel que :

- Le gonflement soit centré au niveau du joint générant la rotation $\mathbf{t} = 0$.
- Le gonflement ait préférentiellement lieu suivant l'axe de rotation. Pour cela, on modifie la proportion d'anisotropie telle que $\mu_y > \mu_x$ et $\mu_y > \mu_z$. On peut considérer par exemple $(\mu_x, \mu_y, \mu_z) = (0, 1, 0)$.
- La distribution du profil de pondération suit la décroissance Gausienne par defaut : $\gamma = \gamma^{\text{defaut}}$.

On obtient ainsi une déformation anisotrope s'adaptant de manière automatique à l'axe de rotation. Le contrôle de celle-ci reste aisé dans la mesure où il consiste uniquement à définir l'influence relative de chacun des trois axes principaux du joint. Un tel effet est présenté en fig. 2.22 c.

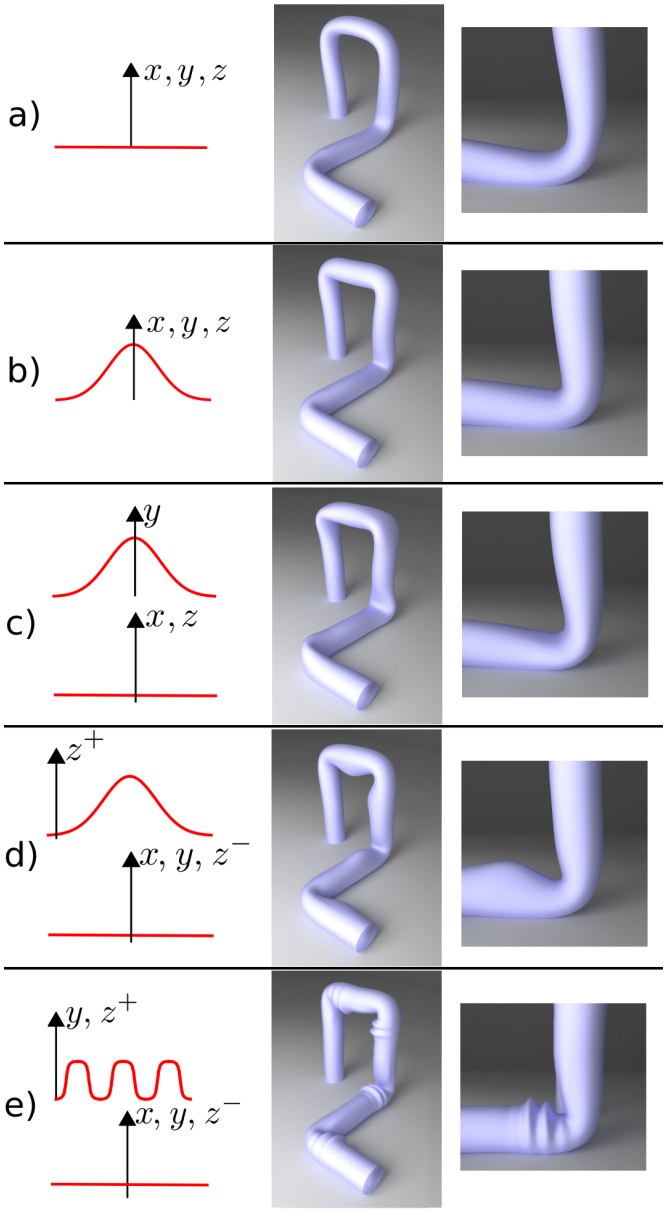


FIGURE 2.22 – *Différents effets possibles lors de la correction de volume. La colonne de gauche représente la forme des modulations A_x, A_y, et A_z. La colonne centrale représente l'effet visualisé sur un cylindre subissant 4 rotations de $\pi/2$ suivant des angles différents, et un zoom sur l'une des articulations est présenté en dernière colonne. a) Déformation par skinning sans compensation de volume à titre de comparaison. b) Inflation isotrope locale. c) Inflation anisotrope locale. d) Inflation unidirectionnelle locale. e) Oscillations.*

Inflation unidirectionnelle locale

Considérons une forme cylindrique dont les mouvements sont générés par des muscles similaires à un biceps. Lorsqu'un tel objet est plié, le muscle à l'origine du mouvement se contracte en générant un gonflement le long de l'os porteur du muscle. De plus, l'inflation n'est pas radiale, mais unidirectionnelle.

Pour modéliser cet effet, nous proposons

- Un gonflement centré au milieu de l'os parent $\mathbf{t} = -L/2\ \mathbf{u}_x$, avec L la longueur de l'os parent.
- Un gonflement n'ayant lieu que dans une unique direction. Cette direction de déformation est donnée par l'axe $\mathbf{u}_z$ qui est à la fois orthogonale à l'axe de l'os et à l'axe de rotation de la pliure. De plus, le gonflement ne se produit que suivant l'orientation positive de cet axe. Nous choisissons donc de modéliser cet effet en modulant la distribution de la correction en volume pour ne déplacer que les sommets k de coordonnées $x_k^L > 0$. Pour cela, on considère
$$A_z(z_k^L) = \max(z_k^L, 0)\ . \tag{2.10}$$
Un tel effet est présenté en fig. 2.22 d.

Oscillations

La compression de formes organiques peut générer des oscillations en forme de bourrelets. Généralement situés sur des parties charnues telles que le ventre, nous pourrons supposer que la déformation est identique à celle d'un gonflement unidirectionnel, mais dont le profil oscille. La modélisation de cet effet est le suivant :

- De manière identique au cas du gonflement de type musculaire, on considérera $\mathbf{t} = -L/2\ \mathbf{u}_x$ et $A_z(z_k^L) = \max(z_k^L, 0)$.
- Nous proposons de plus d'ajouter une partie oscillatoire le long de $\mathbf{u}_x$ correspondant à l'axe de l'os. Nous proposons ainsi de moduler l'amplitude de la pondération γ avec
$$A_x(z_x^L) = \sin^2(\omega\pi x_k^L)\ , \tag{2.11}$$
où ω permet de faire varier la fréquence des plis générés.

Un tel effet est présenté en fig. 2.22 e.

Silhouette quelconque

Les exemples précédents montrent des applications de cas particuliers de formes simples. Dans le cas général, il est souhaitable d'avoir un contrôle intuitif sur l'aspect de la silhouette du gonflement. Nous proposons pour cela de modéliser les fonctions A_x, A_y et A_z à l'aide de courbes unidimensionnelles.

L'utilisateur peut ainsi tracer une courbe de profil suivant l'un des axes privilégiés à l'aide d'outils de dessins dédiés. Ceci permet d'esquisser très rapidement des formes diverses telles que des muscles, de plis de peau, etc. La courbe permet de directement paramétrer les facteurs de modulation de la pondération γ. Nous montrons dans les résultats différentes applications de silhouettes quelconques.

2.5 RÉSULTATS : COMPARAISONS DES DIFFÉRENTES APPROCHES

Nous avons présenté différentes approches pour
- Estimer la modification de volume locale de manière exacte ou approximée
- Appliquer la contrainte de volume de manière exacte ou approximée
- Controler l'apparence finale de la déformation apportée par le respect de la contrainte de manière automatique ou paramétrable manuellement.

Les différentes possibilités peuvent être combinées de manière indépendante. Cependant, nous allons présenter deux cas caractéristiques d'applications de notre méthode sur des maillages de personnages et d'animaux.

1. Le cas d'un skinning rapide à volume constant pour une application typique d'un jeu vidéo. Dans ce cas, le critère principal est de calculer et d'appliquer une correction rapide. Le respect exact de la contrainte n'est pas demandé, et l'on utilisera un paramétrage automatique des pondérations de notre correction.
2. Le cas d'une déformation complexe pour une application typique d'une scène d'un film d'animation, pour laquelle on souhaite principalement un contrôle précis et intuitif de la forme obtenue. On génèrera alors une correction de volume exacte et on paramètrera la pondération de notre correction à l'aide de courbes de silhouettes tracées par l'utilisateur. Le temps de calcul est ici un facteur moins critique.

Dans chacun des cas, nous comparerons les avantages et limitations propres aux méthodes utilisées.

Application au personnage d'un jeu vidéo

Le premier exemple consiste en un éléphant pouvant être utilisé dans le premier cas d'un jeu vidéo. Nous nous inspirons pour cela de modèles réels ou de maquettes aux traits exagérées donnant une impression de volume comme illustré en fig. 2.23.



FIGURE 2.23 – *Modèles d'éléphants dont nous nous inspirons pour modéliser l'effet du contrôle du volume au niveau du ventre. (Les images sont de sources diverses extraites d'internet).*

Le squelette d'animation de notre modèle, sa segmentation en région distincte pour l'estimation de changement de volume local est illustrée en fig. 2.24 gauche. La partie droite de

la fig. 2.24 présente quant à elle les poids de déformations γ calculés automatiquement en fonction des poids de skinning et de la distance au squelette.



FIGURE 2.24 – *Modèle d'éléphant utilisé pour la déformation à volume constant. De gauche à droite : Surface et son squelette d'animation ; Segmentation en région distincte en fonction des poids de skinning ; Mise en place de la pondération automatique au niveau du ventre permettant de localiser la déformation.*

Dans les exemples suivants, nous utilisons la méthode d'estimation rapide du volume de la partie 2.2.1, et la correction présentée en partie 2.3.1. Les deux méthodes étant les plus rapides et chacune n'apportant qu'une approximation de la préservation de volume. La surface est déformée par skinning de manière à ce que le ventre et la trompe subissent une perte locale de volume.

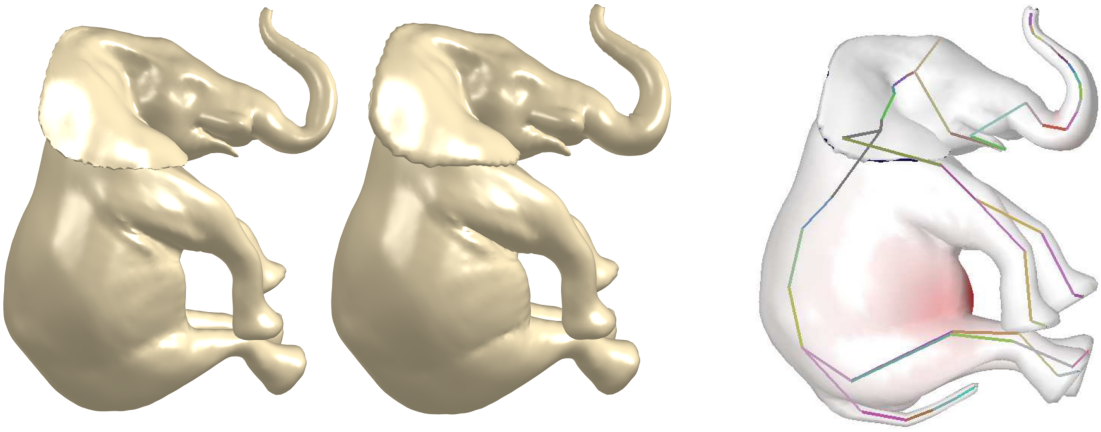


FIGURE 2.25 – *Résultat de la préservation de volume. La figure de gauche correspond à une déformation par skinning classique. Sa perte de volume au niveau du ventre et de la trompe est de 5% du volume initial total. Après correction (milieu), le ventre et la trompe se gonflent localement. L'image de droite représente en couleur les poids de pondérations au niveau du ventre.*

Le résultat obtenu après correction de volume, est illustré en fig. 2.25. Notons que l'amélioration visuelle est également visible au niveau de la trompe de l'éléphant. Le cas spécifique de déformation animée de la trompe est réalisé en fig. 2.26. Bien que la préservation de volume ne

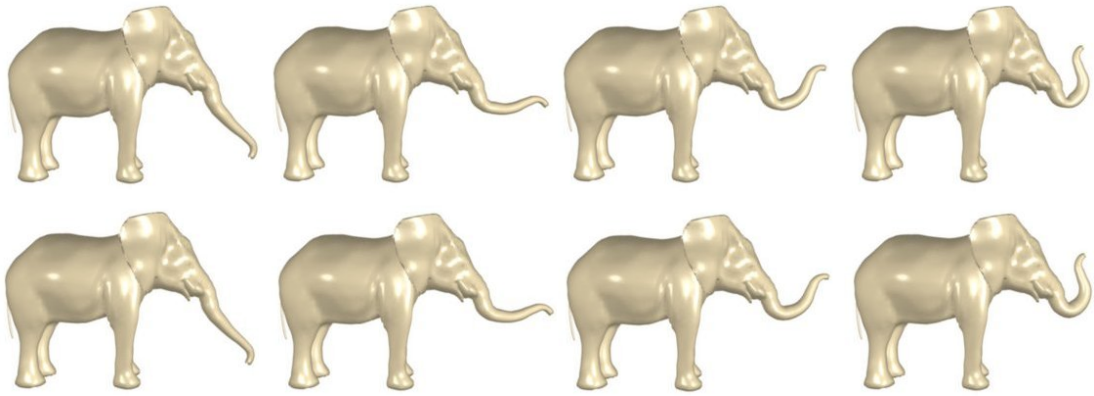


FIGURE 2.26 – *Résultat de la correction de volume sur une animation de trompe pliée. La ligne du haut représente la déformation par skinning, alors que la ligne de bas est obtenue après correction de volume. Notez la partie plate en bas du coude de la pliure dans le cas du skinning standard.*

soit qu'approximée, le résultat visuel reste plausible et donne l'impression d'une préservation locale de celui-ci.

Dans le cas où les différentes régions d'articulation proches peuvent influencer les déformations respectives, nous utilisons l'évaluation exacte du changement local du volume de la partie 2.2.2. Nous articulons alors ce modèle d'éléphant pour réaliser une contraction du ventre doublée d'une déformation des pattes avant dont le rayon d'influence se propage sur le ventre. L'animation d'une telle déformation est illustrée en fig. 2.28, et une comparaison d'une coupe de profil visible en fig. 2.27 montre la déformation apportée par la contrainte de volume.

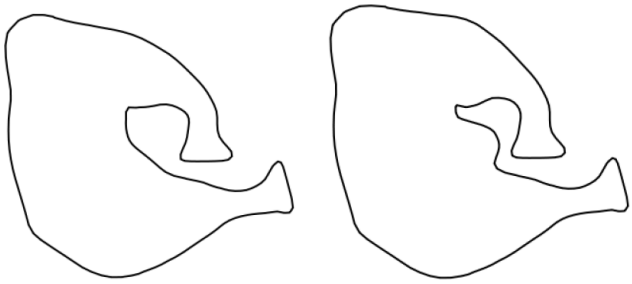


FIGURE 2.27 – *Comparaison du profil de l'éléphant sur une coupe entre une déformation par skinning simple à gauche, et après correction de volume à droite.*

Un autre exemple possible de déformation d'un modèle de jeu vidéo est de considérer l'animation d'un jouet en forme de girafe formé de matière plastique ou en caoutchouc. Nous utilisons cette fois la correction en volume introduite en partie 2.3.2 permettant d'avoir un contrôle manuel précis sur le type de déformation généré. Nous modélisons deux types d'effets :

1. Le cas d'une girafe de type *organique*. Pour cela, nous utilisons les paramètres de déformations par défauts modélisant un gonflement uniforme local centré sur le joint articulé

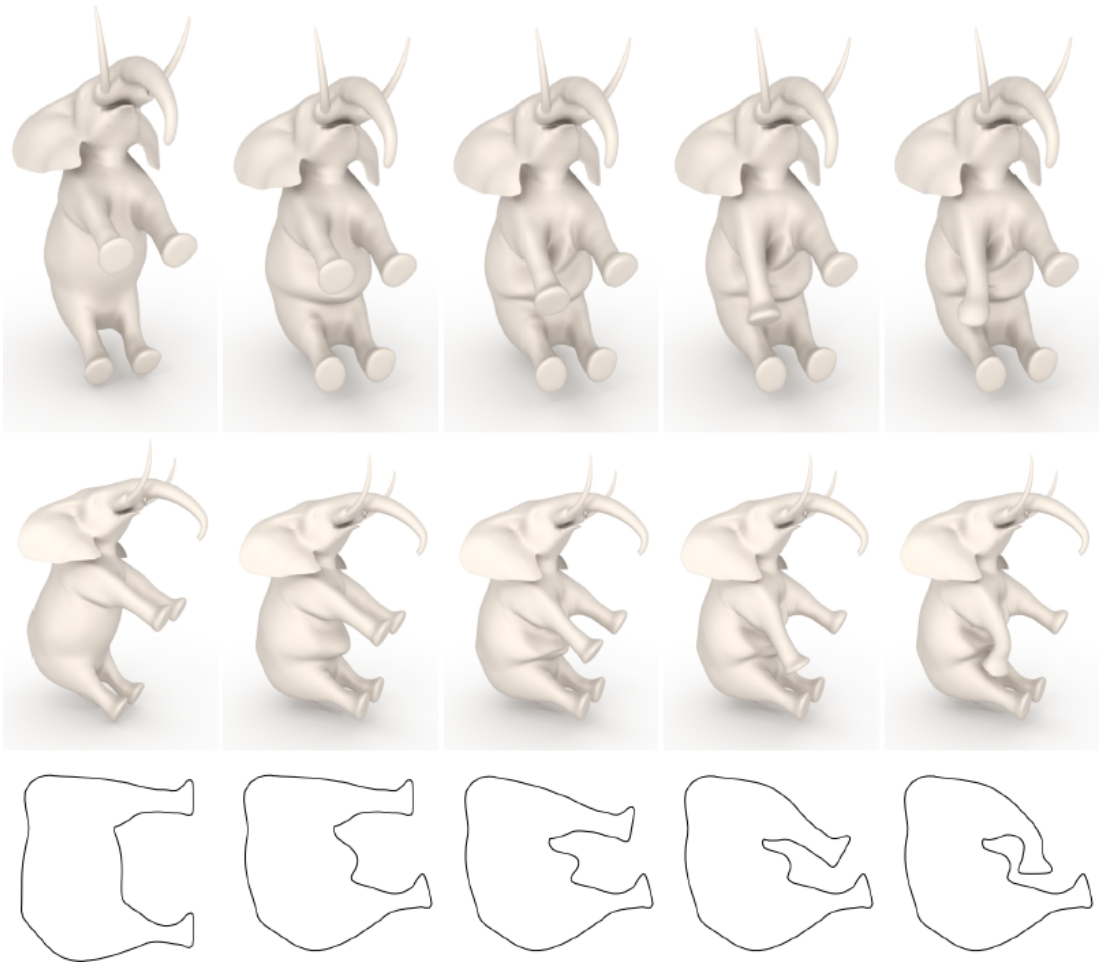


FIGURE 2.28 – *Préservation exacte de volume avec rayons d'influence des corrections se chevauchant. Les deux premières lignes montrent le résultat animé du pliement du ventre et de la patte sous deux angles différents. La ligne du bas représente une coupe de la surface sur un plan.*

comme utilisé pour la fig. 2.22b. Le résultat obtenu sur le modèle de la girafe est visible en fig. 2.29.

2. Le cas d'une girafe formée de matière en caoutchouc. Nous utilisons alors les paramètres de déformations modélisant un gonflement anisotrope le long de l'axe local y avec $(\mu_x, \mu_y, \mu_z) = (0, 1, 0)$ comme décrit précédemment en partie 2.4.2 et visible en fig. 2.22c. Le résultat obtenu sur le modèle de la girafe est visible en fig. 2.30.

Application à la modélisation de détails complexes

Nous mettons désormais en place une correction de volume nécessitant l'introduction d'une déformation secondaire de formes complexes. Pour cela, nous utilisons un outil de dessin vectoriel pour dessiner une courbe de profil pour un os et une direction donnée qui est ensuite utilisé en tant qu'amplitude de pondération pour la pondération γ.

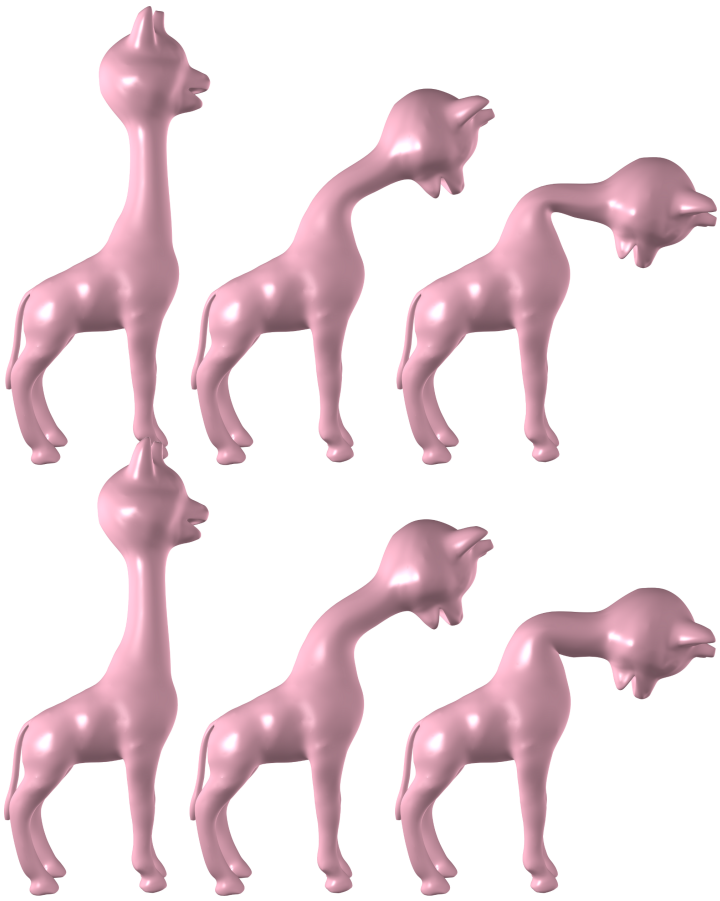


FIGURE 2.29 – *Déformation d'une girafe de type "organique". La ligne du haut représente le résultat d'un skinning standard que l'on peut comparer à la ligne de bas représentant le résultat après notre correction.*

Dans un premier temps, nous étudions le cas du muscle. Nous utilisons pour cela les paramètres utilisés lors de la mise en place du cas d'exemple de la fig. 2.22d. Le résultat obtenu lors de l'application de cette approche sur un bras de personnage est présenté en fig. 2.31

Nous mettons en scène un second exemple d'un éléphant présentant des bourrelets lorsqu'il s'assier. Le principe de l'approche ainsi que la courbe de profil utilisée sont présentés en fig. 2.32. Un résultat d'animation lors de la compression du ventre est également montré en fig. 2.33.

Enfin, une séquence artistique d'un éléphant s'asseyant est présentée dans le teaser de ce chapitre en page 93.



FIGURE 2.30 – *Déformation d'une girafe faite de caoutchouc. De gauche à droite : Modèle original non déformé ; déformation par skinning standard ; résultat après notre correction de volume ; comparaison à un vrai jouet en plastique.*

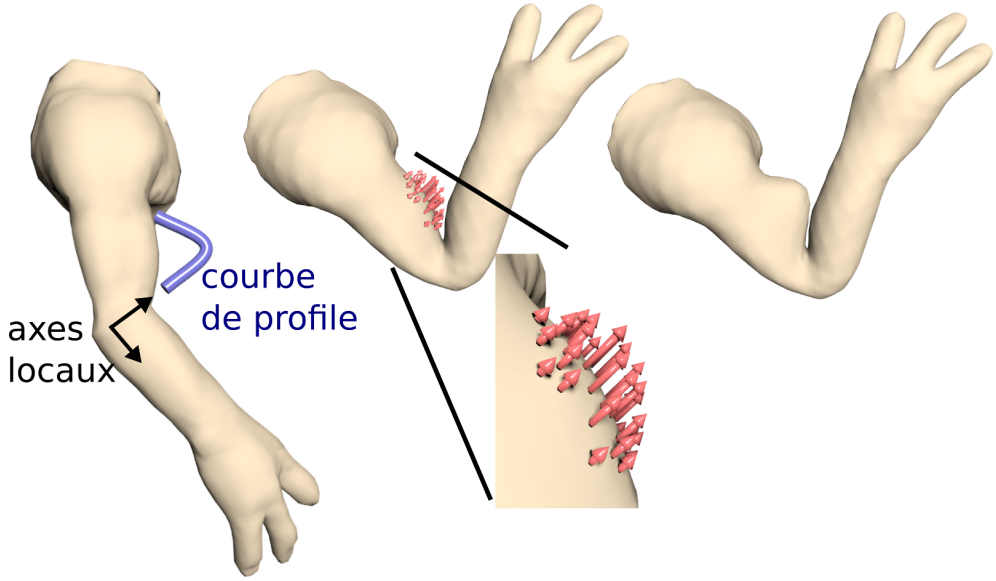


FIGURE 2.31 – *Principe de la modélisation du gonflement musculaire. Une courbe de profil est dessinée et associée à l'os du bras. Lorsque celui-ci se plie, la déformation appliquée permet de réaliser le gonflement du muscle compensant la perte au niveau du coude.*

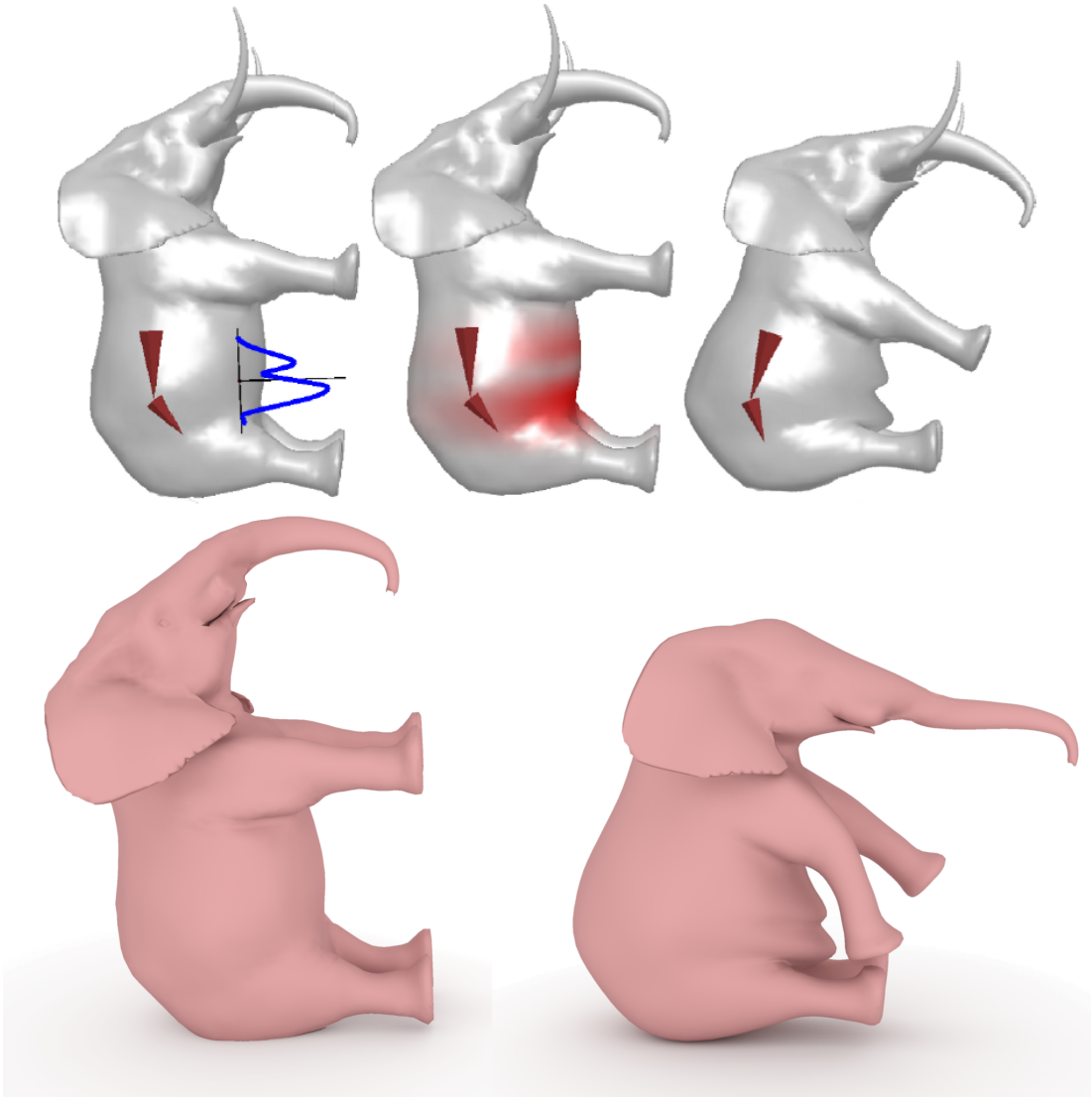


FIGURE 2.32 – *Modélisation de bourrelets sur le ventre d'un éléphant. Ligne du haut : Une courbe de profil est dessinée et associé aux os du ventre, la coloration rouge de la figure du milieu indique les poids γ de pondération résultant de cette courbe. Le résultat finale est montré à droite. Ligne du bas : Rendu final.*

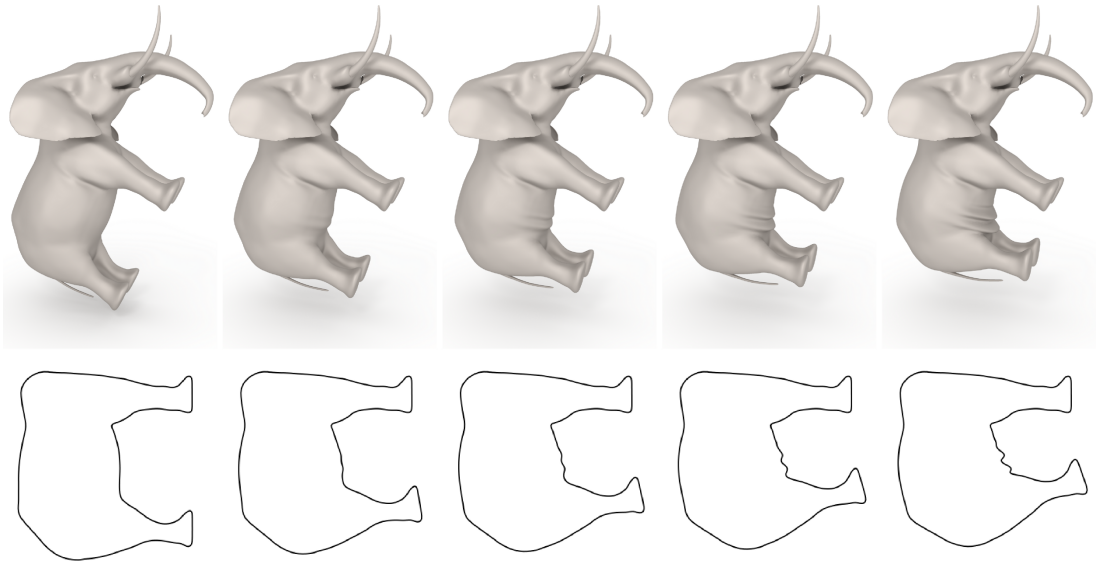


FIGURE 2.33 – *Animation de la création de bourrelets. Ceux-ci s'épaississent au fur et à mesure de la rotation du ventre. Une coupe est visualisée sur la ligne du bas.*

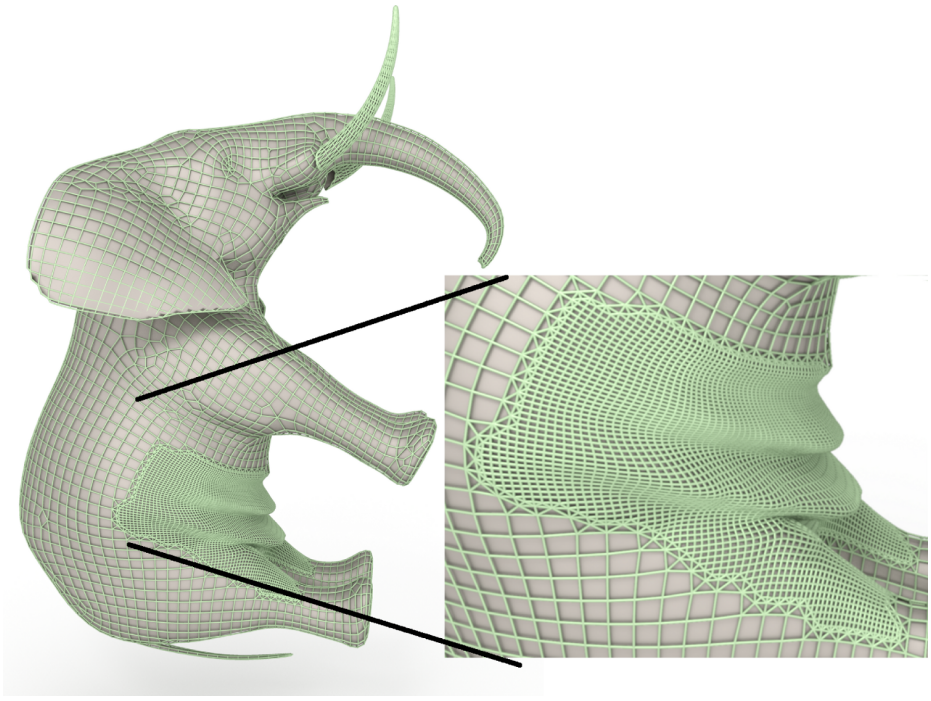


FIGURE 2.34 – *Subdivision locale du maillage s'adaptant aux déformations appliquées.*

Discussion

Temps de calcul Les différents temps de calcul pour les résultats présentés sont résumés dans le tableau 2.1.

modèle		nombre de sommets	temps de calcul (s)
elephant	(fig. 2.25 2.26)	5720	1/25
girafe	(fig. 2.29 2.30)	1667	1/35
éléphant	(fig. 2.28)	5720	1/15
muscle	(fig. 2.31)	838	1/25
éléphant subdivisé	(fig. 2.32)	13439	1/5
éléphant animé	(fig. teaser p.93)	13439	1/2.5

TABLE 2.1 – *Temps de calculs des différents résultats présentés.*

Subdivision locale Les plis complexes définis dans les exemples présentant des oscillations nécessitent l'utilisation d'un maillage suffisamment fin dans les régions détaillées. Or les maillages originaux fournis pour des applications interactives sont généralement à basse résolution. Notons également qu'il est inutile de subdiviser uniformément le maillage lorsque la correction est fortement localisée. Nous proposons ainsi une procédure de subdivision locale s'adaptant à la déformation. Celle-ci se déroule en deux étapes :

1. La déformation par skinning est appliquée sur le maillage grossier. Les polygones potentiellement déformés par correction sont localisés en prenant en compte l'amplitude de la carte de pondération γ pour chaque os entrainant une forte variation de volume ΔV. Les polygones ainsi délimités du maillage non déformé sont alors subdivisés suivant un schéma classique de manière à augmenter la résolution de cette région.
2. L'algorithme de correction de volume est alors appliqué sur le maillage ainsi subdivisé.

Un exemple de telle subdivision est proposé pour le cas des bourrelets du ventre de l'éléphant comme montré en fig. 2.34.

Notons que pour limiter le risque d'artefacts visuels du aux raccordements des polygones, nous élargissons la région à subdiviser au-delà de l'influence de la déformation.

Dépendance à l'ordre de parcours : Comme précisé en partie 2.2.2 et 2.3.2 la méthode de préservation de volume exacte dépend du choix de deux ordres de parcours :

1. L'ordre de parcours du squelette d'animation lors du calcul du changement de volume local.
2. L'ordre de correction suivant x, y puis z lorsque celle-ci s'effectue sur au moins deux de ces axes.

Nous étudions dans ce paragraphe l'impact de ces parcours sur un exemple.

La dépendance du calcul du changement de volume en fonction de l'ordre de parcours du squelette interviens lorsque la correction en volume d'un os au niveau n de la hiérarchie dépend du maillage obtenu à la fin de la correction précédente présente au niveau $n-1$. L'expérience montre cependant que cet effet reste faible pour les maillages considérés.

Soit un cylindre associé à un squelette possédant trois segments pliés aux deux extrémités comme montré en fig. 2.35. Nous appliquons un effet de type *muscle* associé aux deux extrémités en considérant des régions d'influences largement superposées.
Le parcours itératif de l'articulation des os est effectué d'abord de droite à gauche dans un premier cas, puis inversement de gauche à droite dans un second cas. La différence de position du résultat obtenu entre les deux cas est alors calculée sommet par sommet. Celle-ci est imperceptible visuellement et reste inférieure à 2% de l'amplitude de la déformation correspondante. La distribution de l'amplitude de l'erreur sur la surface obtenue est affichée en fig. 2.35

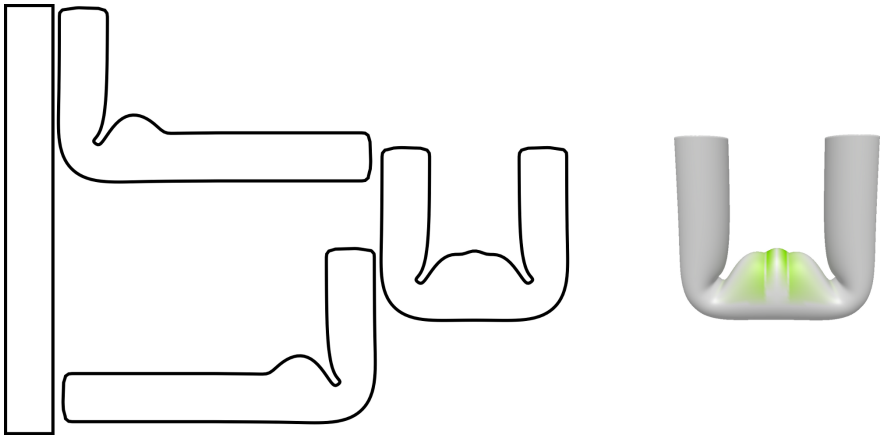


FIGURE 2.35 – *Dépendance de l'ordre de parcours du squelette d'animation. Le tube de gauche représente le cylindre original. Les tubes centraux représentent le premier os articulé dans les deux cas de l'analyse. Le profil résultant après avoir plié les deux articulations est montré à droite. La distribution de la différence entre entre les deux surfaces est encodée en couleur verte sur l'image de droite.*

Nous étudions également l'impact de l'ordonnancement des axes lors de la correction. Un effet de type *muscle* est utilisé avec cette fois $\mu_x = \mu_y = \mu_z = 1/3$. Une première surface est obtenue suivant une correction le long de x, puis y et enfin z. Un second résultat est obtenu en suivant l'ordre inverse. Une coupe visible en fig. 2.36 est réalisée suivant l'axe du cylindre afin de visualiser la différence de profil obtenue. Ici la variation relative maximale s'élève à 5% du rayon du cylindre. Cette variation est visible visuellement, mais les deux résultats restent plausibles et respectent tous deux exactement la contrainte de volume.

2.6 ÉVITEMENT DES AUTO-INTERSECTIONS

La méthode de skinning standard ne garantit pas que le maillage déformé ne subisse pas d'auto-intersection lorsque l'une des articulations est largement déformée. De plus, l'introduction de gonflements pour restaurer le volume peut générer des auto-intersections supplémentaires.

Nous proposons un algorithme permettant de détecter et d'éviter les cas d'auto-intersections simples engendrés par les gonflements de volumes grâce à un partitionnement de l'espace. Pour cela, nous forçons les sommets en auto-collision à se placer sur la surface de contact de leur

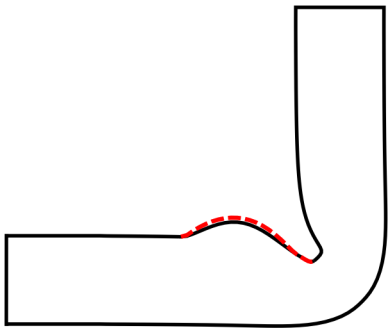


FIGURE 2.36 – *Différence de profil obtenu pour un plis unique dans le cas où la correction en trois étapes est réalisée dans un ordre différent. Les traits noirs représentent le résultat obtenu après correction suivant les axes x puis y et enfin z, alors que les pointillés rouges correspondent au résultat en suivant l'ordre inverse.*

région d'appartenance. Ainsi, une étape supplémentaire de correction de volume permet d'accroitre la déformation associée des régions voisines n'étant pas en contact.

Pour cela, à chaque déformation, nous calculons l'os ayant l'influence maximale suivant la méthode décrite en partie 2.1.1. Pour un sommet donné, lorsque l'os d'influence prépondérante trouvé est différent de celui calculé à la pose de repos, nous en déduisons que celui-ci est entré en collision avec le reste du maillage. Ce sommet est alors déplacé en direction de la surface de contact.

La direction de translation $\mathbf{t}_{ks}$ du sommet k est donnée par le vecteur $I_{sk} - \mathbf{p}_k$, avec s le segment associé initialement au sommet k, et I_{sk} le point le plus proche sur l'os s de la position actuelle $\mathbf{p}_k$. Le déplacement est alors itéré par la relation

$$\mathbf{p}_k^{i+1} = \mathbf{p}_k^i + \Delta s \ \mathbf{t}_{ks}^i \ ,$$

jusqu'à ce que le sommet k regagne sa région originelle (voir fig. 2.37).

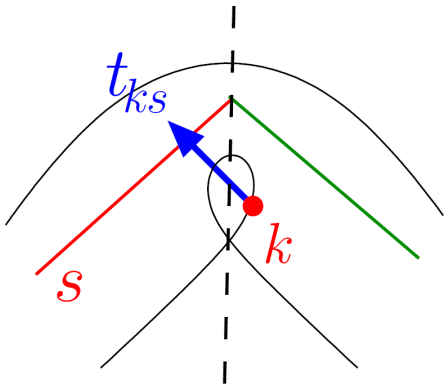


FIGURE 2.37 – *Algorithme de projection du sommet $\mathbf{p}_k$ en auto-collision vers leur os d'origine s. Le vecteur $\mathbf{t}$ est dirigé vers le point le plus proche sur l'os s du sommet k.*

Afin de ne pas réaliser la détection inutilement sur des sommets n'ayant que peu de chances d'entrer en collision, nous ne réalisons ce calcul que pour des sommets situés proche de joints articulés.

Un exemple d'application pour l'évitement d'autocollision dans le cas de doigts pliés est proposé en fig. 2.38.

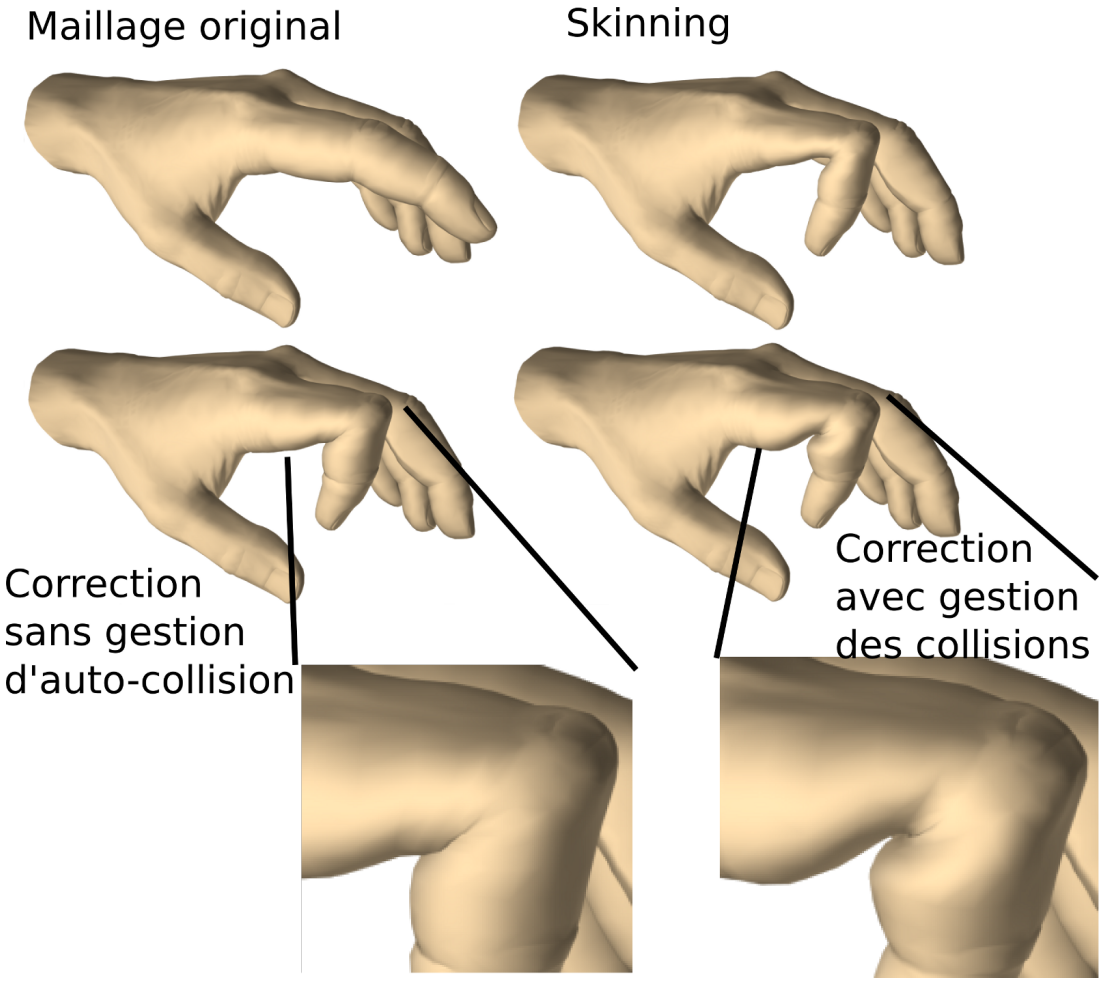


FIGURE 2.38 – *Application de la gestion des auto-collisions dans le cas d'une main. La figure en haut à gauche correspond au modèle original. Le doigt déformé par skinning en haut à droite possède une perte visible de son volume locale. Une correction de volume ne tenant pas compte des intersections vient générer une collision au niveau de la jonction. L'application de l'évitement des auto-collisions viens repousser les sommets en collision et augmenter le gonflement dans la région voisine.*

2.7 INTÉGRATION DANS UNE APPLICATION DE RÉALITÉ AUGMENTÉE PRÉEXISTANTE : LE HAND NAVIGATOR

Le *Hand Navigator* est un périphérique externe permettent de contrôler interactivement les degrés de liberté d'une main virtuelle et de ses doigts. Son rôle est de permettre la manipulation d'objets 3D en se rapprochant de la sensation d'une manipulation réelle. Ce périphérique est développé sous forme de prototype par Chardonnet *et al.* [CALC09b, CALC09a] et illustré en fig. 2.39.

La manipulation d'objets virtuels prend toute sa dimension si le modèle virtuel associé peut se déformer de manière réaliste et interactive. Nous proposons pour cela de modéliser un objet déformable à l'aide d'un skinning à volume constant qui réagira à la présence d'une main virtuelle définie par une surface skinnée dans l'espace. En particulier, nous nous intéressons à la possibilité de plier les différentes parties du modèle virtuel sous l'action d'une rotation de la main. Ainsi que la possibilité de tirer ou compresser l'objet lors de la translation de celle-ci.

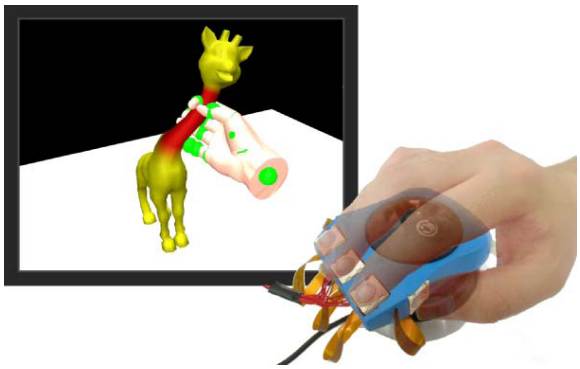


FIGURE 2.39 – *Dispositif du Hand-Navigator développé par Chardonnet et al. utilisé pour la déformation d'un objet virtuel. Image issue de [CAL+10].*

Le cas particulier de la compression ou de l'extension lié à la translation d'un os est une application moins classique du skinning (au contraire de la rotation), car son application traditionnelle génère l'extension ou la disparition irréaliste de matière comme illustrée en fig. 2.40 gauche. L'utilisation d'une contrainte de volume constant permet ici de modéliser le gonflement ou le dégonflement lié à l'extension d'un os.

Par rapport aux résultats présentés précédemment, l'interfaçage avec la main virtuelle guidée par le *Hand Navigator* possède des particularitées propres :

- La déformation des joints (rotation et translation) n'est plus contrôlée directement, mais de manière indirecte par le déplacement de la main virtuelle.
- Le squelette ainsi que le maillage doit être adaptable pour prendre en compte interactivement les zones de contacts avec la main virtuelle.
- La main génère des déformations par itérations successives. Pour éviter de stocker l'historique de l'ensemble des déformations, la pose *initiale* du skinning sera considéré comme celle issue de la déformation précédente.

Nous détaillons les solutions utilisées pour ces différents points dans la suite.

2.7.1 Articulation par transferts de déformations

La main virtuelle doit venir guider les déformations du personnage. Pour cela, nous définissons une correspondance entre l'articulation de la main en contact avec le personnage et l'os du personnage le plus proche.

Soit $\mathbf{p}_{inter}$ la position du contact entre la main et l'objet. Soit T_{main} la matrice de déformation associée à l'os de la main le plus proche de $\mathbf{p}_{inter}$. On applique alors la déformation T_{main} à l'os le plus proche du personnage. La déformation de l'objet suit alors automatiquement les mouvements de la main en contact de manière naturelle comme illustré en fig. 2.40 droite.

Notons que cette manipulation permet de modéliser des contacts simples entre une main et l'objet. Le cas de la préemption à plusieurs doigts permettant le pincement ou l'ajout de contraintes supplémentaires n'est pas traité dans le cas présent.

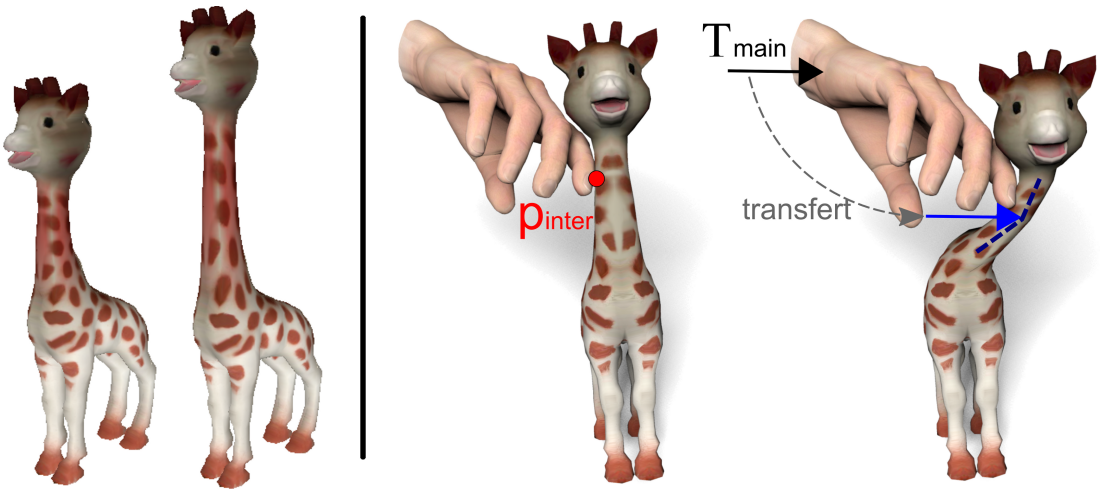


FIGURE 2.40 – *Gauche : Elongation d'un os du coup du modèle générant une création irréaliste de matière. Droite : Transfert de déformation entre la matrice de déformation de la main et l'os le plus proche du point d'intersection de la main avec la surface.*

Une fois le transfert de déformation appliqué, il est possible d'obtenir des allongements et rétrécissements d'os traduit par un gonflement/dégonflement local de la surface. Un exemple de tel résultat est montré en fig. 2.41.

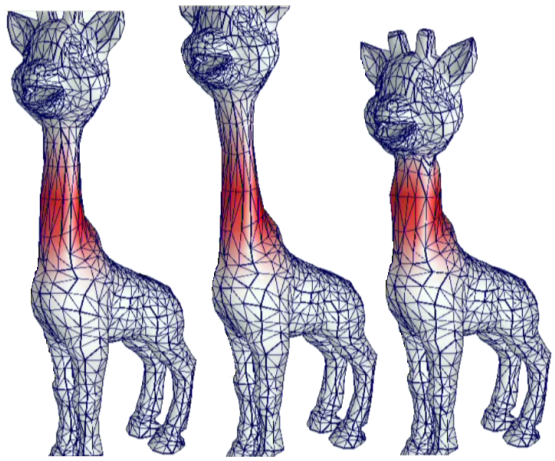


FIGURE 2.41 – *De gauche à droite : Maillage original ; après extension de l'os du coup par translation générant une diminution du rayon de celui-ci ; après réduction de l'os du coup par translation générant un gonflement local.*

2.7.2 Déformations locales par squelette et maillage dynamique

L'utilisateur ayant un contrôle indirect sur l'articulation des os du personnage n'est à priori pas sensé avoir connaissance de l'organisation du squelette d'animation. De plus, il peut potentiellement vouloir déformer localement des zones qui sont initialement uniquement liées rigidement à un os unique. Pour modéliser ce type d'action locale, il est nécessaire d'**adapter dynamiquement** le squelette et le maillage du personnage afin d'ajuster les degrés de liberté aux zones de contacts locales.

Supposons que la main vienne en contact de l'objet à déformer dans le but de le pousser ou tirer localement de la matière. Lorsque qu'un contact avec la main est défini à la position $\mathbf{p}_{\text{inter}}$, nous générons un os supplémentaire entre $\mathbf{p}_{\text{inter}}$ et $\mathbf{p}_m$. $\mathbf{p}_m$ étant défini comme la position la plus proche du squelette d'animation original à $\mathbf{p}_{\text{inter}}$.

L'os contenant $\mathbf{p}_m$ est alors scindé en deux, et la branche $(\mathbf{p}_{\text{inter}}, \mathbf{p}_m)$ est ajoutée à la hiérarchie du squelette. Les poids de skinning sont recalculés localement.

Pour permettre des déformations d'étirement successives, nous proposons également de localement subdiviser le maillage dans la région dépendante du nouvel os ajouté. Ainsi, des déformations locales peuvent être appliquées indépendamment de la finesse du maillage original. Un exemple de déformation locale d'une partie de la tête de la girafe est présenté en figure 2.42.

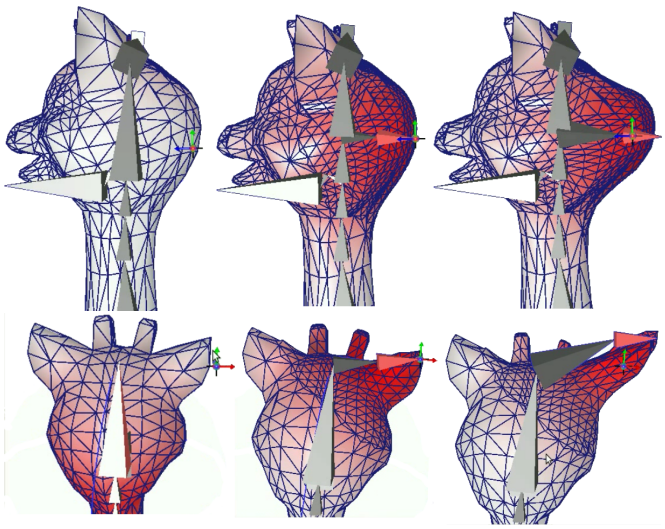


FIGURE 2.42 – *Déformations locales appliquées à une région de la girafe. La ligne du haut présente une déformation de la surface de l'arrière du crâne, alors que la ligne du bas illustre la déformation de l'oreille. Pour chaque cas l'image de gauche représente le maillage et le squelette original, l'image du milieu représente le nouveau squelette après détection du contact ainsi que le maillage localement subdivisé. Enfin, la figure de droite représente le résultat obtenu après déformation.*

2.7.3 Préservation de volume itérative

L'utilisateur doit pouvoir manipuler et déformer l'objet de manière itérative. Le calcul de volume se référant toujours à une position initiale il existe deux approches possibles :

1. Considérer la déformation totale entre l'objet original et l'objet courant.
2. Considérer la dernière déformation temporellement appliquée. L'objet de référence étant itérativement modifié à chaque déformation.

Le premier cas permet d'assurer de retrouver l'objet original en appliquant la déformation inverse sur les os. Par contre, le suivi de l'ensemble des déformations sur toutes les articulations à chaque étape rend le calcul couteux et difficilement gérable dans le cas d'ajout dynamiques d'os.

Le second cas permet quant à lui de ne traiter que localement le cas de l'articulation déformé à un instant courant. Le calcul s'en trouve accéléré. De plus, il possède l'avantage de pouvoir adapter la surface considérée comme référence à des traitements quelconques qui ne sont pas forcément modélisable par skinning. Pour cette raison nous choisissons d'implémenter cette approche.

Cependant, cette méthode possède également deux inconvénients :

1. Dans le cas d'une préservation de volume non exacte, les erreurs sur la contrainte viendront d'additionner itérativement.
2. L'application de la transformation inverse ne garantit pas de pouvoir revenir à l'état original. Par exemple, dans le cas d'une torsion d'un os d'un angle de π radians, des sommets se retrouvent spatialement confondues. Cette configuration étant considérée pour

la déformation suivante, il ne sera plus possible par la suite de les dissocier. Une telle configuration à éviter est illustrée en fig. 2.43.

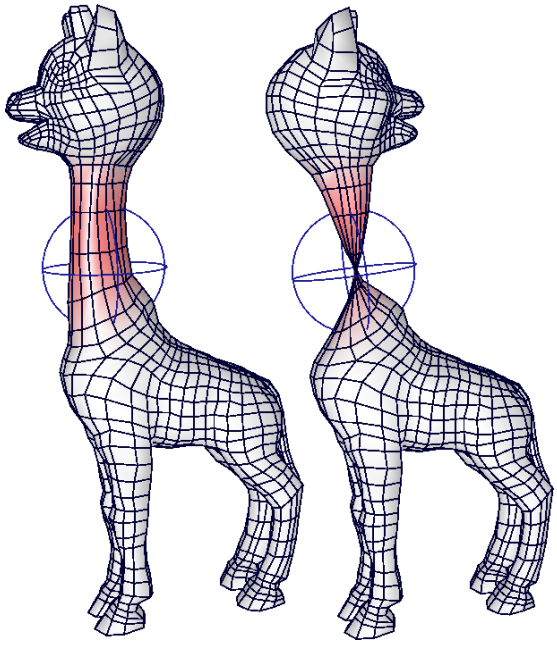


FIGURE 2.43 – *Transformation de l'espace non bijective suivant une rotation de π radians. Une fois que la figure de droite est obtenue et considérée comme nouvelle pose de référence, il n'est plus possible de séparer les sommets du coup qui ont convergé vers une unique position spatiale.*

BILAN DE L'INTÉGRATION DANS UN ENVIRONNEMENT DE MANIPULATION INTERACTIVE

La déformation par skinning à volume constant à pu être intégrée de manière concluante dans une application de manipulation interactive. La combinaison entre un dispositif de réalité augmenté pré-existant et la déformation plausible d'un personnage virtuel est une première étape vers une manipulation plus naturelle d'objets complexes.

Cette intégration a fait l'objet d'une présentation lors d'un salon [GIF09] ou des personnes non-expertes ont pu manipuler la scène virtuelle comme le montre la fig. 2.44.

Pour permettre une manipulation réellement réalistes d'objets ce type d'approche pourrait faire l'objet de diverses améliorations futures. Nous avons pu noter que la prise en compte de contraintes multiples des doigts de la main serait un plus. D'autres méthodes de déformations pourraient également être étudiées dans le cas d'interactions difficilement modélisées par l'articulation d'un squelette interne. Pour cela, les approches implicites ou la modélisation de méthode de sculpture représentent une piste d'avenir prometteuse.



FIGURE 2.44 – *Manipulation du prototype de déformation interactive par le publique d'un personnage virtuel lors d'un salon. Image issue de [CAL+10].*

2.8 EXTENSION AUX CONTRAINTES D'AIRE CONSTANTE

L'étude de déformation de la surface d'objets plus précise montre que des contraintes structurelles peuvent imposer que ceux-ci se déforment à aire ou même à isométrie préservée. L'isométrie étant traitée dans les deux chapitres suivants, nous nous intéressons au cas particulier de la préservation d'aire sur la surface. Les matériaux très légèrement élastiques tels que la peau vont satisfaire à de telles contraintes de manière approximées. Par exemple, un personnage pliant sa main va voir se générer des plis de peaux de par une contrainte de non-compression pouvant se traduire par une préservation d'aire.

Nous montrons dans cette partie qu'il est possible d'incorporer des contraintes de préservation d'aire tout en assurant une déformation à volume fixé.

Principe de la préservation de volume et d'aire

Considérons le cas de la préservation d'aire globale pondérée suivant les méthodes présentées précédemment. Nous considérons une déformation dans la direction normale à la surface telle que présentée en partie 2.3.1.

En réutilisant les notations précédentes, nous pouvons écrire la nouvelle équation que nous souhaitons désormais résoudre

$$\left\{ \begin{array}{ll} \min & \|\boldsymbol{\rho}\|_\gamma^2 \\ \text{contraint à} & V(\mathbf{P}+\boldsymbol{\rho}\mathbf{N}) = V_{\text{target}} \\ \text{contraint à} & A(\mathbf{P}+\boldsymbol{\rho}\mathbf{N}) = A_{\text{target}} \ , \end{array} \right. \tag{2.12}$$

avec l'aire A de la surface donnée dans le cas d'une triangulation par

$$A(\mathbf{P}) = \frac{1}{2}\sum_{i=1}^{N_f} \|(\mathbf{p}_{b_i} - \mathbf{p}_{a_i}) \times (\mathbf{p}_{c_i} - \mathbf{p}_{a_i})\| \ .$$

La contrainte d'aire est non linéaire. Similairement à l'expression du volume, nous linéarisons cette expression suivant une faible déformation par ρ. On peut ainsi écrire

$$A(\mathbf{P}+\boldsymbol{\rho}\mathbf{N}) \simeq A(\mathbf{P}) + < \boldsymbol{\rho}\mathbf{N}, \nabla A > \ ,$$

où $\nabla A \in \mathbb{R}^{3N_s}$ correspond au vecteur des dérivées partielles de A concaténées par rapport à chaque coordonnées spatiale. On notera également similairement au cas du volume $\Delta A = A_{\text{target}} - A(\mathbf{P})$.

Introduisons les notations vectorielles

$$\left\{ \begin{array}{l} \mathbf{N}_V =< \mathbf{N}, \nabla V > \in \mathbb{R}^{3N_s} \\ \mathbf{N}_A =< \mathbf{N}, \nabla A > \in \mathbb{R}^{3N_s} \end{array} \right.$$

On peut alors montrer (voir annexe B.4) que la solution analytique satisfaisante de manière approchée aux contraintes d'aire et de volumes est donnée par :

$$\begin{aligned} \forall k \in [\![1, N_s]\!] \ , \quad \rho_k \quad = \quad & \frac{\gamma_k}{\|\mathbf{N}_V\|^2_{1/\gamma}\|\mathbf{N}_A\|^2_{1/\gamma} - < \mathbf{N}_V, \mathbf{N}_A >^2_{1/\gamma}} \\ & \Big[\Big(\Delta V \|\mathbf{N}_V\|^2_{1/\gamma} - \Delta A < \mathbf{N}_V, \mathbf{N}_A >_{1/\gamma}\Big) \mathbf{N}_V(\mathbf{p}_k) \\ + \quad & \Big(\Delta A \|\mathbf{N}_A\|^2_{1/\gamma} - \Delta V < \mathbf{N}_V, \mathbf{N}_A >_{1/\gamma}\Big) \mathbf{N}_A(\mathbf{p}_k)\Big] \ . \end{aligned} \tag{2.13}$$

Résultat

En appliquant ce cas sur un exemple de bras articulé, nous pouvons constater l'effet de l'ajout de la contrainte de préservation d'aire en fig. 2.45. Nous pouvons constater que bien que les corrections ne soient qu'approximées, au bout de 4 étapes successives d'applications de l'eq. (2.12), nous obtenons une erreur en volume et en aire négligeable. Nous notons également visuellement que la préservation d'aire permet de lisser le résultat final. La contrainte de volume stricte pouvant engendrer de légères oscillations dues à la triangulation (comme expliqué en partie 2.2). La préservation d'aire force quant à elle a garder des triangles de taille homogène à leur valeur initiale et empêche les extensions locales.

	sans correction	contrainte de volume	volume + aire (1 itération)	volume+aire (4 itérations)
erreur en volume.	3.3%	<1e-5%	<1e-5%	<1e-5%
erreur en aire.	1.3%	0.7%	0.2%	<1e-5%

TABLE 2.2 – *Tableau d'erreur en pourcentage de volume et d'aire en fonction de la correction mise en place pour le cas de la fig. 2.45.*

Discussion et synthèse

Nous avons pu voir qu'il était possible d'introduire de manière similaire d'autres contraintes géométriques en supplément de la préservation de volume.

En appliquant la solution analytique approchée moins de 5 fois, nous obtenons une déformation préservant à la fois le volume et l'aire en des régions pondérées localement.

Notons que pour l'instant, la préservation d'aire est réalisée de manière globale et ne permet donc pas la mise en place de déformations locales. Dans le cas où nous souhaiterions simuler

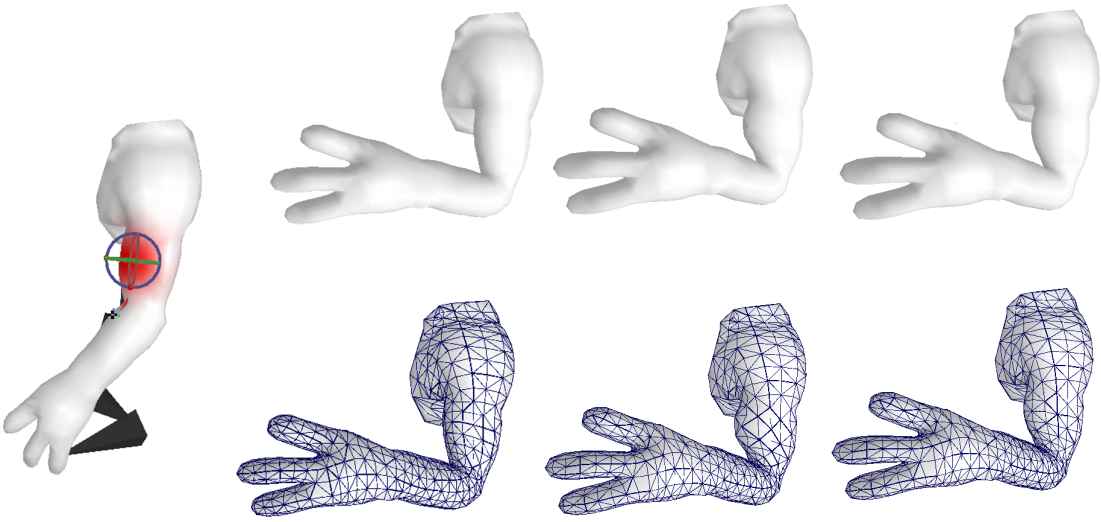


FIGURE 2.45 – *Résultat de l'application de la contrainte de volume et d'aire au niveau d'un bras articulé. De gauche à droite : Pose de repos avec placement manuel de la pondération γ au niveau du biceps permettant de simuler un effet de muscle ; Déformation par skinning standard sans corrections, on peut noter l'absence d'effet de muscles ; Résultat après correction de volume uniquement, on peut noter le gonflement du muscle ainsi que des oscillations dues à la triangulation ; préservation de volume et d'aire (après 4 itérations), l'effet de biceps est toujours présent, et de plus la surface reste lisse.*

l'effet de plissement de peau de par la préservation d'aire, il serait nécessaire de réaliser la préservation et la correction localement. Il sera ainsi possible d'étendre ce travail en modifiant l'équation à résoudre pour considérer des sous régions de quelques triangles sur lesquels l'aire pourra être préservée localement.

BILAN DE L'ANIMATION À VOLUME CONTRÔLÉ

Nous avons présenté dans cette partie une méthode de déformation de personnages et créatures virtuelles. La méthode consiste à ajouter un post-processing sur une déformation par skinning standard permettant de contrôler localement la valeur du volume. Nous avons mis en place une génération automatique de poids de skinning, l'estimation locale du changement de volume, puis la déformation géométrique permettant de contrôler celui-ci.

Pour chacune de ces étapes, nous avons proposé deux modes d'estimations se prêtant à deux cas d'applications différentes :

- Le cas des jeux vidéos nécessitant des calculs rapides et automatiques, mais limités à un contrôle de volume approximé.
- Le cas de la génération de contenu 3D demandant un contrôle de haut niveau sur la forme obtenue pour lequel nous proposons une approche robuste et exacte associée à un contrôle utilisateur par courbes de profils.

Pour garantir l'applicabilité de cette méthode à des déformations d'objets complexes, nous avons proposé une méthode d'évitement de collisions dues aux gonflements locaux, ainsi

qu'une subdivision locale permettant d'incorporer des détails géométriques nouveaux tels que des bourrelets.

Nous avons montré que cette approche a pu s'intégrer et être testée dans le contexte d'une déformation interactive d'objets virtuels en apportant un gain de plausibilité sur le résultat visuel.

Finalement, nous avons montré que la méthode pouvait aisément s'étendre à des contraintes géométriques différentes telles que la préservation d'aire.

La génération de plis locaux tels que ceux présents sur la peau ou les vêtements nécessitent cependant la prise en compte de comportements très locaux s'intégrant difficilement dans une déformation articulée guidée par un squelette d'animation. Le problème principal concernant la gestion de la localisation et de la direction de ces plis sur la géométrie actuelle.

Pour répondre à ces problèmes, nous proposons au chapitre suivant une solution générale de création de plis basée sur la préservation des longueurs pouvant s'appliquer à de l'animation de vêtement tout autant qu'à une déformation par squelette.

CHAPITRE

3

ANIMATION DE SURFACES QUASI-INEXTENSIBLES : APPLICATION AUX PLIS DE VÊTEMENTS.



En dépit de multiples années de recherche, la modélisation efficace et réaliste de déformation de tissu reste complexe. Bien que les méthodes de déformation physique permettent de modéliser des effets dynamiques tout en tenant compte des collisions, la simulation précise de plis demande un temps de calcul important. L'animation de détail de plis reste cependant un facteur essentiel pour rendre crédible les séquences de jeux vidéos (fig. 3.1) et de production cinématographique (fig. 1 droite en p. 15), mais le temps de calcul nécessaire à la simulation empêche souvent d'utiliser ces méthodes.



FIGURE 3.1 – *L'ajout de plis sur les animations de vêtements dans les jeux vidéos récent permet de rendre ceux-ci visuellement plus dynamique et plus réaliste. Image de gauche issue du jeu vidéo* Star Wars : The Force Unleashed, LucasArts*, notez les plis sur la jambe du personnage. Image de droite provenant du jeu vidéo* Virtua Tennis 4, Sega*. Les plis sur la robe du personnage rendent la scène plus dynamique.*

En effet, les jeux vidéos nécessitent des algorithmes de déformation temps-réel et la simulation numérique détaillée n'est pas applicable dans de tels contextes. En production, bien que le temps de calcul ne soit pas la problématique principale, l'absence de manipulation directe de la forme finale obtenue oblige les artistes à lancer plusieurs simulations avec différents paramètres à de nombreuses reprises afin se rapprocher des résultats visuels qu'ils souhaiteraient obtenir. Au final, l'investissement en temps machine et en manipulation manuelle de paramètres donne à l'approche par simulation physique brute un coût pouvant se révéler rapidement prohibitif.

Contexte de notre approche

Une méthode alternative à la simulation pour générer des plis de tissus a été proposée pour une application en production par Cutler *et al.* [CGW+05] (voir partie 1.4.3) et consiste à améliorer une simulation grossière à l'aide de plis ajoutés et paramétrés manuellement. Cependant, cette approche possède deux inconvénients principaux :

1. Le placement et paramétrage manuel de la structure de plis demande un investissement temporel conséquent et une grande expertise à la fois technique et artistique.

2. Le design manuel peut uniquement être réalisé à certains pas de temps, alors que les plis se déforment continument au cours du temps, entremêlant alors des comportements dynamiques complexes de mélanges et de séparations.

Comme dans les travaux de Cutler *et al.*, notre méthode permet de conserver une étape de simulation, effectuée en temps réel sur un maillage grossier, afin de calculer la dynamique d'une animation de tissus. Par contre, notre approche est une alternative à cette définition manuelle. Elle permet de générer des plis plausibles lors d'un post-traitement comme illustré en fig. 3.2 par une méthode entièrement automatique. De plus, tout en assurant la plausibilité du résultat, cette méthode permet l'apparition et la disparition dynamique des plis. L'approche peut s'appliquer de manière automatique sur des animations préexistantes de tissus tels que des simulations masses-ressorts ou du skinning.

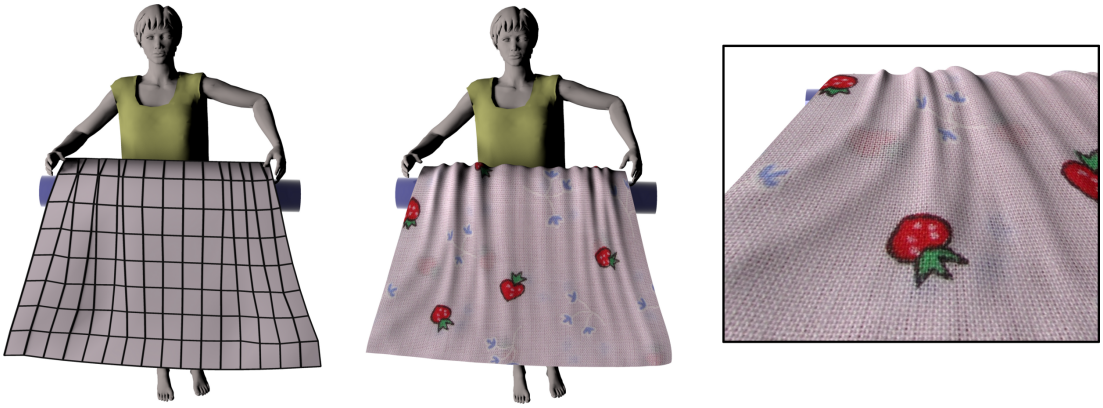


FIGURE 3.2 – *Entrée-sortie de la méthode décrite. Gauche : Une surface à basse résolution issue d'une simulation rapide présente des compressions locales. Droite : Notre méthode ajoute des plis géométriques dans les régions compressées et suivant une orientation adaptée afin d'améliorer automatiquement la plausibilité du résultat.*

Nous observons dans un premier temps que la formation des est fortement liée à la développabilité des tissus, ou en d'autres termes, à la quasi-isométrie entre la surface modélisant un tissu animé et son patron 2D dans sa pose de repos [PZB+09].
Ainsi, étant donné une simulation à basse résolution, nous utilisons l'information d'élongation issue de la comparaison entre la surface déformée par rapport à son patron pour guider le placement géométrique des plis à ajouter. Pour obtenir une animation temporelle de plis cohérents, la position et l'amplitude des plis sont modifiées de manière continue au cours du temps. La géométrie 3D des plis est générée en utilisant un mécanisme basé sur des déformeurs implicites pouvant modéliser des formes de plis complexes, des variations de rayons et de profondeur de la géométrie du plis, ainsi que la gestion automatique des transitions entre des plis se mélangeants et simplement adjacents.

Les contributions principales de ce chapitre ont été présentées dans les publication suivantes : [RPC+10], [RHC10].

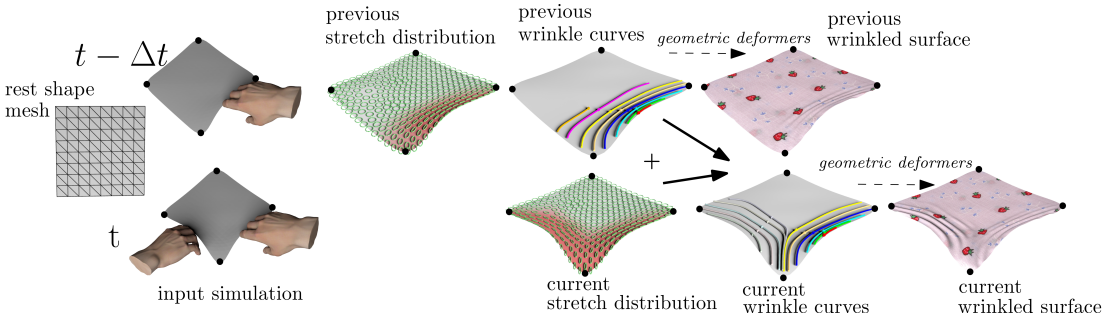


FIGURE 3.3 – *Principe de notre méthode. La ligne du haut représente les étapes à l'instant précédent $t - \Delta t$ alors la ligne du bas représente celles à l'instant courant t. À partir d'un maillage de repos donné, nous générons une déformation rapide par simulation ou skinning introduisant des compressions non réalistes. L'analyse des tenseurs d'élongations permet de générer des lignes de plis dans les régions compressées. Pour obtenir une animation cohérente au cours du temps, les lignes de plis de l'instant précédent sont utilisées pour définir celles de l'instant courant l'instant courant. Enfin, l'approche par déformeur implicite permet de générer une géométrie réaliste de plis de tissus.*

Principe général de notre approche

Étant donné que la quantité de matière de tissu reste constante au cours du temps, les plis de tissus se forment dans les régions où la simulation numérique résulte en des compressions. Ces plis se forment alors de manière perpendiculaire aux directions de compression principale.

S'appuyant sur cette observation, notre méthode analyse les tenseurs de compressions sur le maillage grossier par rapport à la pose de repos. Nous générons alors des courbes de plis dans les régions où les tenseurs indiquent une compression. Pour cela, nous traçons les courbes intégrales le long des directions principales de compressions définies par le champ de tenseurs. Afin de maintenir la cohérence temporelle entre les plis lors d'animation dynamique, notre méthode n'est pas appliquée indépendamment à chaque pas de temps, mais utilise l'information des tenseurs pour propager les courbes d'un pas de temps à l'autre ainsi que pour créer ou supprimer des courbes lorsque cela est nécessaire.

Les plis réels se forment de manière à préserver l'isométrie par rapport au patron original. De plus, leurs amplitudes et leurs rayons dépendent du type de tissu manipulé ainsi que de son épaisseur. Pour modéliser ce type d'effet, notre méthode utilise les tenseurs d'élongation pour déterminer l'épaisseur et la profondeur locale des plis de manière à ce que l'ajout de ceux-ci permette de réduire le défaut d'isométrie par rapport à la pose de repos. Étant donné l'ensemble des courbes de plis calculés ainsi que leur profondeur et amplitude associée, notre méthode génère des plis géométriques plausibles à l'aide d'une nouvelle méthode basée sur des déformeurs implicites.

Une vue générale des différentes étapes décrites dans le document est présentée en fig. 3.3.

Dans l'ensemble de la description de cette méthode, nous paramétrons le type de tissu, et donc de plis par la variable $R_{\min}$ correspondant au rayon de courbure minimale d'un pli. Cette variable au sens intuitif pour un utilisateur non spécialiste dépend directement de l'épaisseur et de la raideur du tissu.

3.1 GÉNÉRATION STATIQUE DE LIGNES DE PLIS

Nous décrivons tout d'abord l'extraction de courbes unidimensionnelles symbolisant la localisation des plis à partir d'un maillage déformé et de sa forme au repos associée. La méthode décrite peut-être utilisée telle quelle pour ajouter des plis après une modélisation statique telle que pour l'habillage de mannequins. Le cas de l'animation temporelle étant discutée en section 3.2

Dans la majorité des cas, la pose de repos considérée est un patron planaire. Dans le cas où ce patron 2D n'est pas connu, nous le générons à partir de la surface de référence non déformée que nous segmentons et paramétrons en parties quasi-développables [HLS07] comme illustré en fig. 3.4. Nous utilisons l'algorithme ABF++ décrit par Sheffer *et al.* [SLMB05] afin d'obtenir un bon compromis entre préservations de longueurs et d'angles. Afin de maintenir la cohérence entre les chaque couture de la surface, nous contraignons la paramétrisation à préserver les longueurs. Cette étape de paramétrisation définit un système de coordonnées homogènes pour les triangles du maillage modélisant la surface de référence afin d'améliorer l'efficacité de l'algorithme du tracé de courbes de plis.

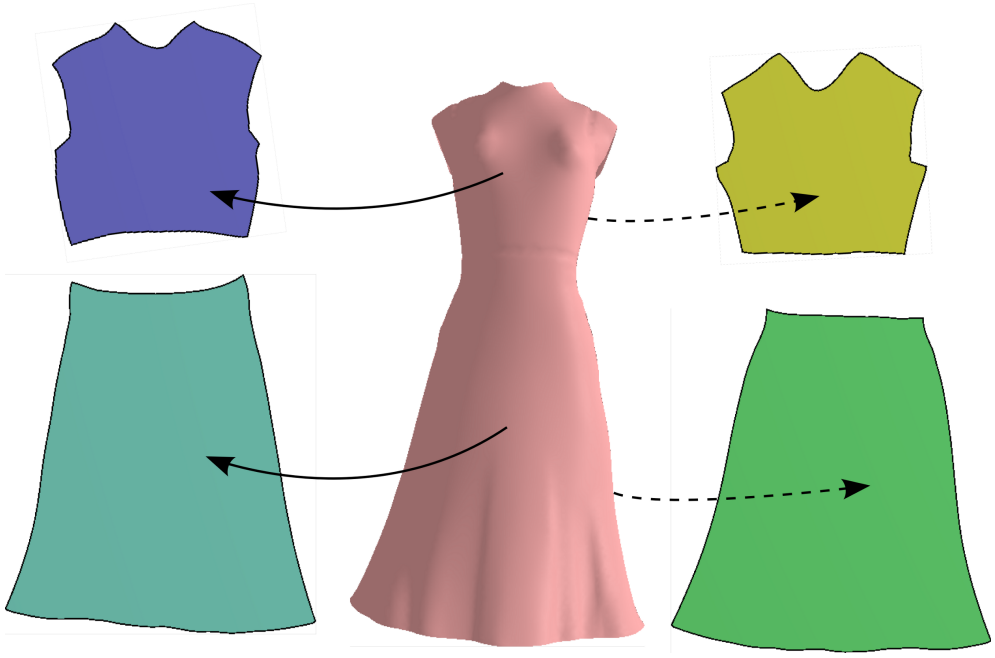


FIGURE 3.4 – *Exemple de paramétrisation d'un vêtements en 4 parties. Chaque partie 2D forme le patron de sa correspondance 3D.*

3.1.1 Champ de vecteurs issu des tenseurs d'élongation

Comme indiqué précédemment, les plis de tissus sont orientés orthogonalement à la direction principale de compression. Nous cherchons ainsi un *champ de vecteurs de plis* défini sur la surface du tissu tel que sont amplitude soit reliée à la quantité de compression du modèle à basse résolution, et dont la direction soit orthogonale à la direction locale de la compression.

Formulation du tenseur d'élongation par triangle

Désignons par S le maillage 3D représentant le tissu, et $\overline{S}$ le maillage représentant la pose de repos correspondante comme illustré en fig. 3.5 La déformation affine 2D d'un triangle de S vers son correspondant dans $\overline{S}$ peut être exprimée par une matrice 2×2 T.

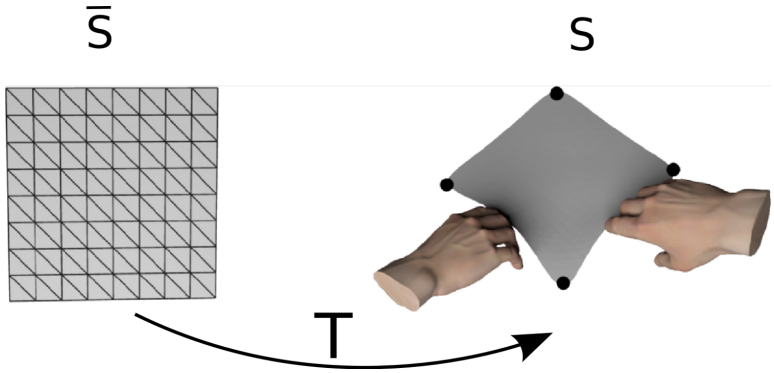


FIGURE 3.5 – *Déformation du patron 2D $\overline{S}$ vers une surface 3D S.*

Appellons $(\mathbf{u}_1, \mathbf{u}_2)$ et $(\overline{\mathbf{u}}_1, \overline{\mathbf{u}}_2)$ les cotés 2D des triangles dans la base locale de S et $\overline{S}$ (voir fig. 3.6). On peut alors définir

$$\mathrm{T} = [\mathbf{u}_1, \mathbf{u}_2][\overline{\mathbf{u}}_1, \overline{\mathbf{u}}_2]^{-1} .$$

La déformation T est isométrique (aucune élongation ni compression) si et seulement si

$$\mathrm{T}^T = \mathrm{T}^{-1} . \tag{3.1}$$

Nous retrouvons alors la version discrête de la définition d'isométrie basée sur la première forme fondamentale donnée en partie 1.4.3 eq.(1.32). Soit U la matrice de taille 2×2 définie par

$$\mathrm{U} = \sqrt{\mathrm{T}^T \mathrm{T}} .$$

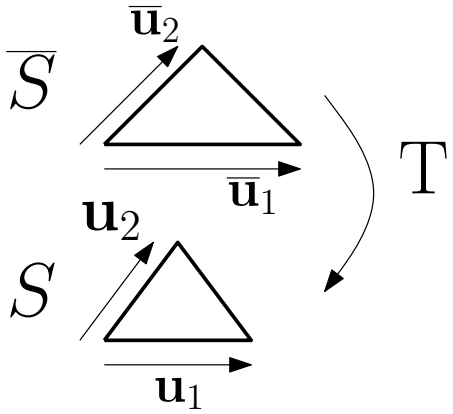


FIGURE 3.6 – *Déformation d'un triangle dans la base locale.*

U est appelée *tenseur d'élongation* et correspond à la racine carré du *tenseur de déformation de Cauchy Green* en mécanique des milieux continus. Notons que cette représentation est similaire à celle donnée par Sander *et al.* [SSGH01].

Cette matrice représente la mesure de la compression et de l'élongation d'un triangle par rapport à sa position de référence indépendamment de son orientation spatiale. En supposons qu'aucun triangle de S où $\overline{S}$ ne soit dégénéré, U est par définition une matrice symétrique définie positive. Elle peut donc être diagonalisée sous la forme (voir fig. 3.7)

$$\mathrm{U} = \lambda_1 \mathbf{e}_1 \mathbf{e}_1^T + \lambda_2 \mathbf{e}_2 \mathbf{e}_2^T \,, \tag{3.2}$$

où $\lambda_1 \leq \lambda_2$ sont les valeurs propres réelles positives. $\mathbf{e}_1, \mathbf{e}_2$ représentent les vecteurs propres et définissent les directions principales d'élongation et/ou de compression. Une illustration d'une telle visualisation pour différentes déformations est montrée en fig. 3.8.

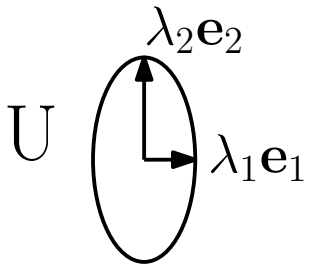


FIGURE 3.7 – *Ellipsoide liée à la décomposition en éléments propres de* U.

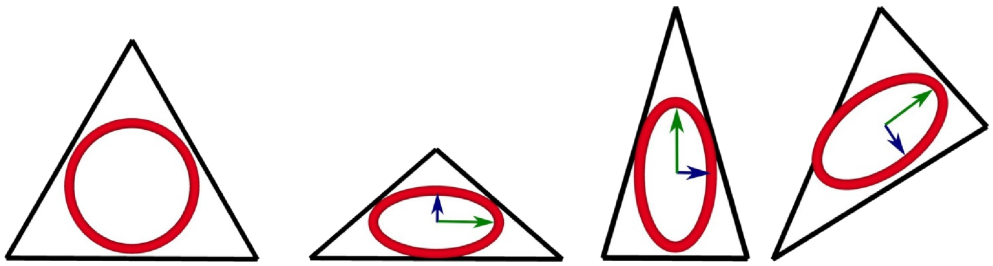


FIGURE 3.8 – *Gauche : triangle original. Les trois figures suivantes représentent le même triangle une fois déformé, ainsi que la représentation sous forme d'ellipsoïde des éléments propres du tenseur d'élongation* U.

Estimation d'un champ de tenseurs continu

Pour extraire une carte de plis à partir de l'information tensorielle par triangle, nous définissons dans un premier temps un champ de tenseurs continu -par sommet- décrivant des directions de compressions variant continument sur la surface.

Interpolation des tenseurs En utilisant le système de coordonnées défini par la paramétrisation, nous définissons le tenseur d'élongation à un sommet k par

$$\mathrm{U}_k = \bigoplus_{j \in \mathcal{V}_k} \omega_j \mathrm{U}_j \,, \tag{3.3}$$

avec $\mathcal{V}_k$ désignant les indices des triangles incidents au sommet k, et $\omega_j = \mathcal{A}_j / \sum_{k \in \mathcal{V}_j} \mathcal{A}_k$ étant la pondération des triangles voisins basée sur leurs aires respectives.

Nous notons que l'utilisation d'une interpolation linéaire triviale des composantes du tenseur aboutit à la dissipation de l'information d'anisotropie.
Pour mieux préserver cette information, nous utilisons un schéma d'interpolation sur un invariant affine d'une métrique riemannienne défini dans l'espace des tenseurs. Cette approche fut introduite par Pennec *et al.* [PFA06].
Dans l'eq.(3.3), le signe $\oplus$ désigne alors la moyenne pondérée calculée itérativement dans

l'espace des tenseurs, et U_k est alors obtenue par application de la formule suivante jusqu'à convergence

$$\mathrm{U}_k = \mathrm{U}_k^{1/2} \exp\left(\sum_j \omega_j \log\left(\mathrm{U}_k^{-1/2}\,\mathrm{U}_j\,\mathrm{U}_k^{-1/2}\right)\right)\mathrm{U}^{1/2}\,. \tag{3.4}$$

Cette approche permet de générer des valeurs propres variant continument, et, plus important, dont les directions des vecteurs propres tournent de manière continue tout en minimisant la dissipation de l'information d'anisotropie initiale comme illustrée en fig. 3.9. Nous définissons alors le champs de tenseur continu U sur la surface en utilisant les coordonnées barycentriques et l'opérateur d'interpolation tensoriel $\oplus$.

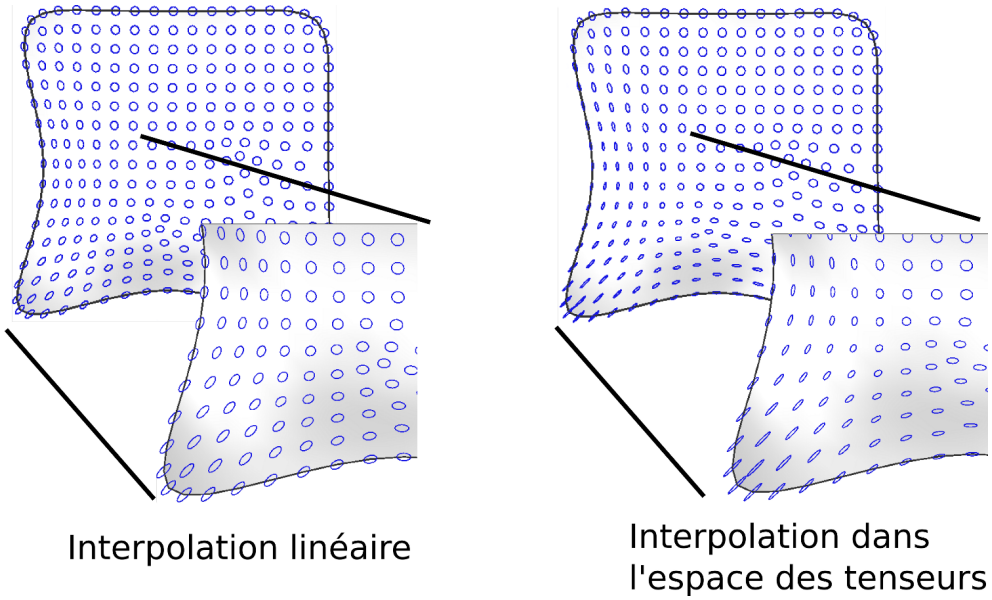


FIGURE 3.9

Définition du champ de vecteurs de plis Dans l'optique de générer des plis, nous nous intéressons uniquement aux régions où le maillage original a subi une compression, c'est-à-dire les lieux où la valeur propre la plus faible λ_1 est inférieur à 1. Ainsi, nous définissons un *champ de vecteurs de plis* $\mathbf{v}$ pour l'ensemble des positions de la surface tel que

$$\mathbf{v} = \max(1-\lambda_1, 0)\,\mathbf{e}_2\,. \tag{3.5}$$

L'intensité $\|\mathbf{v}\| := v$ reflète le taux de compression exprimé par unité de longueur sur la surface lorsque celle-ci est compressée. La direction de $\mathbf{v}$ étant orthogonale à la direction de compression principale.

3.1.2 Tracé de courbes de plis

Nous souhaitons générer efficacement des courbes symbolisant des lignes de plis afin de modéliser de manière plausible le comportement de la surface de tissu en réponse à une compression mesurée sur le modèle à basse résolution.

Rôle des courbes de plis

Nous souhaitons calculer des courbes qui capturent à la fois la forme et l'amplitude du champ de vecteurs de compression tout en échantillonnant convenablement les régions compressées.

Dans cette optique, notre approche consiste à tracer des courbes dans les régions fortement compressées tout en imposant une distance minimale entre les courbes.
Cependant, les plis réels peuvent en fait se rapprocher fortement ou même se mélanger dans certaines régions, violant alors cette contrainte de distance minimale entre courbes. Pour prendre en compte ce phénomène, nous proposons de générer les courbes une à une en débutant aux sommets associés aux compressions maximales tout en contraignant les nouvelles courbes à débuter sur des sommets situés suffisamment loin des courbes déjà tracées.
En forçant cette contrainte de distance minimale uniquement sur les sommets sources des courbes, il devient alors possible de générer des courbes pouvant se rendre arbitrairement proches les unes des autres. Les déformeurs implicites que nous définissons en partie 3.3.2 permettent de traiter le cas de plis se mélangeant de manière robuste.

Expression des courbes de plis

Nous définissons les courbes de plis par des courbes paramétrées

$$\overline{\gamma} : [0,1] \rightarrow \overline{S}\,, \tag{3.6}$$

générées en tant que courbes intégrales issues du *champ de vecteurs de plis* $\mathbf{v}$ sur la surface paramétrique de repos 2D.

L'équation des courbes peut s'exprimer sous forme d'équation aux dérivées ordinaires

$$\begin{cases} \overline{\gamma}'(u) = \pm\mathbf{v}\left(\overline{\gamma}(u)\right) \\ \overline{\gamma}(0) = \overline{\mathbf{s}} \\ \|\mathbf{v}\left(\overline{\gamma}(u)\right)\| \geq 1 - \lambda_{\max}\,, \end{cases} \tag{3.7}$$

où $\overline{\mathbf{s}}$ représente une position de départ de la courbe $\overline{\gamma}$, et $(1-\lambda_{\max}) \in [0,1]$ représente le taux de compression minimal caractérisant les régions pour lesquels des plis vont être générés. En pratique, on utilisera généralement $0.8 \leq \lambda_{\max} \leq\simeq 0.95$.

Initialisation des courbes de plis

Pour initialiser le tracé des courbes, nous ordonnons les sommets du maillage dans une file de priorité ordonnée suivant leur taux de compression associé. Les courbes débutent ainsi des sommets associés à la compression la plus grande. On appelle l'origine de cette courbe son *générateur*.

Pour tracer les courbes une à une, nous procédons de la manière suivante :

1. **Integration** : Étant donné que $\mathbf{v}$ possède un degré de continuité suffisant d'après notre expression du tenseur d'élongation U, nous traçons une courbe lisse démarrant de la position du générateur issu de la file de priorité en utilisant une méthode d'intégration de type Euler explicite.
   Comme le champ $\mathbf{v}$ n'est pas orienté, nous propageons la courbe dans les deux directions à partir du générateur. Pour avancer dans une direction, nous vérifions à chaque pas d'intégration spatial que l'angle entre les directions précédente et courante reste inférieur

à $\pi/2$ radians.
Le processus d'intégration s'arrête lorsque le taux de compression $\|\mathbf{v}\|$ devient plus petit que $1 - \lambda_{\max}$, où lorsque l'on atteint le bord de la surface.

2. **Sélection de la position du générateur** : La position du générateur de la courbe suivante est choisie en sélectionnant le sommet suivant dans la file de priorité tout en éliminant les positions trop proches des courbes déjà existantes. Cette distance minimale peut être choisie automatiquement comme étant égale au double du rayon de courbure minimale $R_{\min}$. En effet deux courbes définies à une distance plus proche viendraient générer des plis se mélangeant.

Le processus se termine finalement lorsque la file de priorité est vide.

Une fois tracées, les courbes 2D $\overline{\gamma}$ peuvent être transférées sur la surface déformée 3D à partir du mapping défini par des coordonnées barycentriques d'un maillage à l'autre.
Les courbes 3D désignées par γ indiquent alors le lieu et la direction où les plis géométriques doivent être ajoutés afin de compenser la compression locale. De plus, le taux de compression défini le long de chaque courbe peut être utilisé pour contrôler la forme des plis comme discuté en partie 3.3.2.

Un exemple de telles courbes définies à partir de la carte des tenseurs est illustré en fig. 3.10

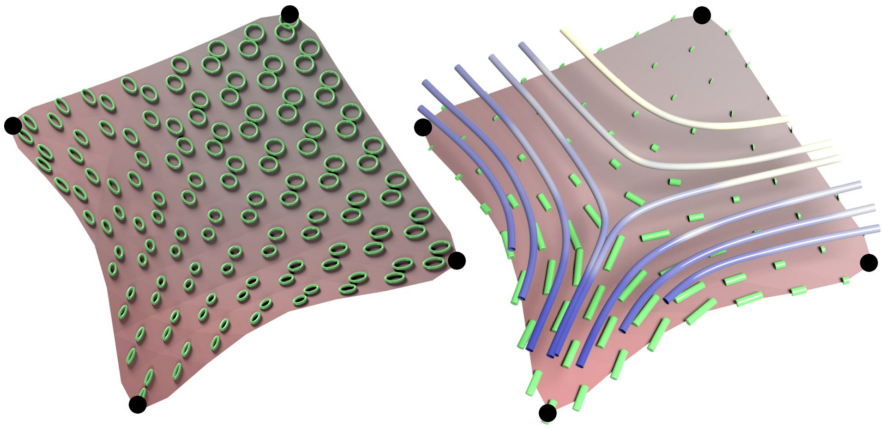


FIGURE 3.10 – *Exemple de carte de tenseurs et de courbes de plis formées par les lignes intégrales sur un exemple de rectangle compressé sur le milieu de ces deux bords.*

3.2 ANIMATION DYNAMIQUE DES COURBES DE PLIS

Les plis de tissus réels se forment et évoluent de manière continue au cours du temps. À l'opposé, l'utilisation en l'état de l'algorithme décrit dans la section précédente aboutissait à la génération de plis indépendants pour chaque pas de temps. Cela peut générer des séquences animées pour lesquels les plis vont apparaitres et disparaitres de manière brutale, faisant place à des plis similaires mais voisins.

Popa *et al.* [PZB+09] se basent sur une entrée vidéo en combinaison avec une étape d'optimisation spatio-temporelle couteuse afin d'établir la continuité temporelle. Dans le cas présent,

nous ne disposons pas de l'apport d'une vidéo et souhaitons définir une solution rapide permettant d'assurer la continuité temporelle.
Dans cette optique, nous nous basons sur une approche particulaire pour effectuer un lissage temporel permettant d'obtenir l'effet souhaité.

3.2.1 Propagation des positions génératrices des courbes de plis

Premièrement, nous constatons que la continuité de l'animation de la surface fournie en entrée génère un *champ de vecteur de plis* variant continument d'un pas de temps à l'autre.
Nous pouvons donc supposer que la plupart des courbes de plis générées à un instant donné soient également présentes à l'instant suivant. Ce pli peut cependant avoir potentiellement glissé sur la surface et avoir changé légèrement de forme.

Principe du glissement des plis

Pour modéliser l'effet de glissement potentiel d'une courbe de pli, nous initialisons celles-ci à l'instant actuel à partir de la position initiale des courbes de l'instant précédent. On rappelle que notre génération statique place les lignes de plis aux endroits de compression maximale, et trace ces lignes le long des courbes intégrales du champ de vecteur.

Pour étendre cet algorithme au cas de courbes dynamiques, nous souhaitons attirer les courbes de plis vers les nouveaux maxima de compressions. Dans l'optique de rester efficace, nous ne cherchons pas les maximums locaux le long des courbes entières, mais uniquement pour la position du générateur.
Ainsi, nous définissons une méthode particulaire permettant d'attirer les générateurs vers les maximums locaux de la compression. Nous décrivons l'algorithme ci-après.

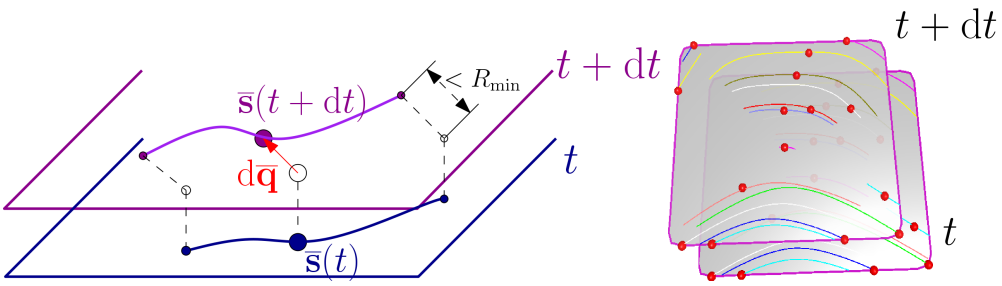


FIGURE 3.11 – *Gauche : Principe du glissement de plis particulaires dans l'espace 2D du patron. La courbe bleue correspond à la courbe de pli à l'instant actuel. Pour déterminer la position de la courbe mauve à l'instant suivant, le générateur $\overline{s}$ est projeté à l'instant $t+\mathrm{d}t$ puis translaté par le vecteur $\mathrm{d}\overline{\mathbf{q}}$. La courbe mauve est alors construite par intégration en partant de cette nouvelle position génératrice. Droite : L'évolution de courbes de plis sur le patron entre deux instants supperposés.*

Algorithme d'attraction des générateurs

La méthode d'animation particulaire ordonne les positions génératrices des courbes en fonction de leur taux de compression associé.
Pour chaque position d'origine $\overline{\mathbf{s}}$ positionnée en $\overline{\mathbf{s}}(t)$ au temps t, sa nouvelle position au temps

$t + \mathrm{d}t$ est donnée par $\bar{\mathbf{s}}(t) + \mathrm{d}\bar{\mathbf{q}}$, où $\mathrm{d}\bar{\mathbf{q}}$ est un petit déplacement du patron 2D modélisant le glissement du plis.
Nous utilisons $\mathrm{d}\bar{\mathbf{q}} \neq 0$ si la compression au point d'origine décroit par rapport à l'instant précédent. Dans ce cas, le vecteur de déplacement $\mathrm{d}\bar{\mathbf{q}}$ est orienté dans la direction du gradient de plus forte compression $\nabla\|\mathbf{v}\|$ dans le but de se diriger vers le maximum local de compression.

Pour déterminer le pas de déplacement optimal, nous utilisons une recherche linéaire le long de la direction du gradient, tout en limitant le déplacement maximal autorisé à $R_{\min}$.
Ce choix empêche donc les plis de glisser de plus de la moitié de leur largeur lors d'un pas de temps, et permet un résultat visuel plausible.

Pour éviter que plusieurs plis ne viennent converger en direction d'un même maximum local, nous contraignons chaque générateur à rester à une distance minimale de $2R_{\min}$ des autres courbes. Ensuite, nous traçons les courbes au pas de temps actuel en utilisant l'intégration le long des lignes de champs décrites en section 3.1.2. Enfin, nous observons le déplacement des extrémités de chaque courbe par rapport à l'instant précédent. Si celui-ci est plus grand que $R_{\min}$, la courbe est alors rétrécie ou allongée en suivant la forme de la courbe au pas de temps précédent afin d'éviter que l'intégration n'engendre des discontinuités temporelles importantes. La fig. 3.11 illustre ce principe de glissement de plis sur 2 pas de temps.

3.2.2 Ajout et suppression de courbes

Nous ajoutons également à cette procédure d'animation un mécanisme permettant de faire disparaitre ou d'ajouter des plis.

Lorsque le taux de compression local n'est plus suffisamment important malgré l'étape de glissement, le générateur et sa courbe associée sont supprimés à partir de cette étape.
Finalement, après que l'ensemble de générateurs préexistants aient été traités, de nouveaux générateurs sont définis dans les régions nouvellement compressées en appliquant la même procédure que dans le cas statique. Un exemple d'ajout et de suppression de plis est présenté en fig. 3.12.

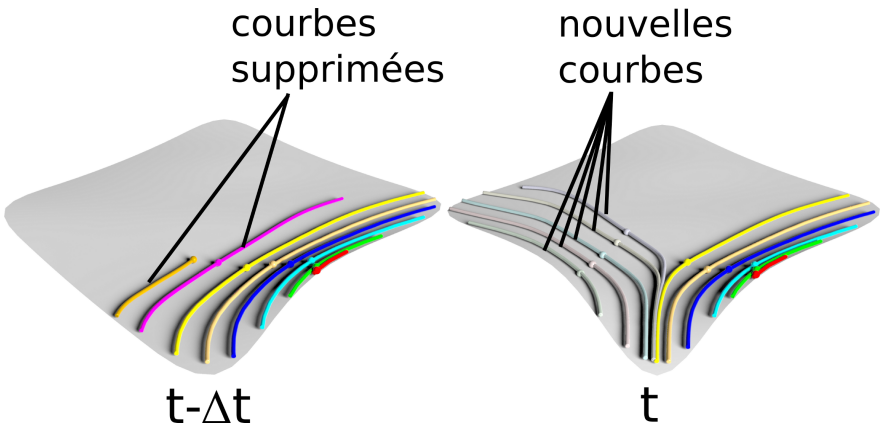


FIGURE 3.12 – *Exemple d'ajout et de suppression de plis entre différents pas de temps (on considère ici un pas de temps artificiellement grand, à des fins d'illustrations.)*

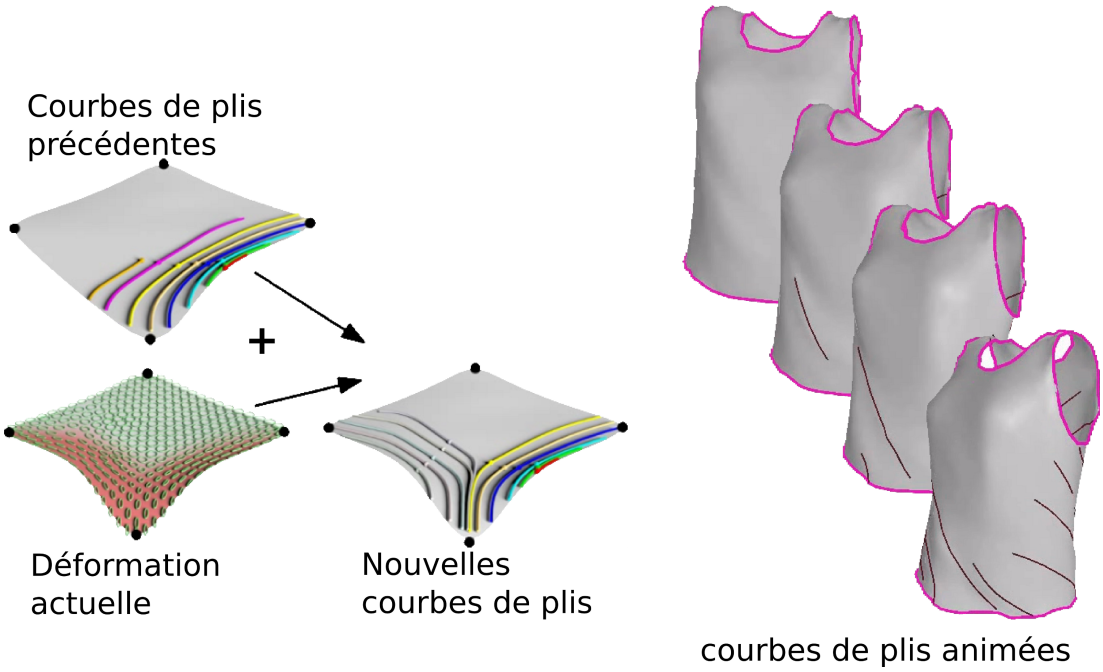


FIGURE 3.13 – *Principe résumé de l'animation dynamique des courbes de plis, et application de celle-ci sur un T-shirt en mouvement. Les courbes de plis se créent dans les régions compressées, se déplacent et disparaissent de manière continue en fonction du mouvement de la surface.*

3.2.3 Lissage temporel

L'algorithme que nous venons de décrire permet l'animation plausible de plis de tissus au cours du temps. En particulier, l'amplitude locale de compression augmente et décroit de manière suffisamment douce. Nous n'observons ainsi pas discontinuités brutales lorsque les courbes de plis sont ajoutées ou supprimées.
Cependant, bien que les trajectoires temporelles des générateurs soient continues, la construction intégrale des courbes est calculée de manière indépendante pour chaque pas de temps. Celle-ci peut ainsi présenter des discontinuités temporelles lorsque la courbe s'éloigne de son générateur.

Pour contraindre la continuité temporelle, nous lissons les trajectoires intégrales dans le domaine du temps.
Supposons que les courbes de plis soient spatialement paramétrées de manière homogène suivant le paramètre u. Une courbe donnée peut ainsi être considérée comme une fonction de l'espace et du temps $\overline{\gamma}(u,t)$.
Nous appliquons alors un filtre passe-bas unidimensionnel sur chaque courbe de variable temporelle $\overline{\gamma}(u = \text{const}, t)$ dans le but d'assurer des transitions lisses entre chaque pas de temps.

L'application de l'algorithme ainsi décrit permet la mise en place d'animation de courbes de plis dynamiques sur la surface. L'application de cette approche sur l'animation d'un T-shirt est illustrée en fig. 3.13.

3.3 GÉNÉRATION GÉOMÉTRIQUE DES PLIS PAR DÉFORMEURS IMPLICITES

Une fois que les courbes de plis ont été calculées pour tous les pas de temps, l'étape finale consiste à générer la géométrie des plis sur la surface.
La forme des plis, c'est-à-dire leur largeur et leur profondeur, reflète à la fois la quantité de compression appliquée sur la surface, l'épaisseur du tissu ainsi que sa structure interne. Par exemple, la compression de tissu en soie génèrera de nombreux plis fins alors que la compression d'un morceau de cuir formera des plis beaucoup plus larges.

3.3.1 Paramètres des plis

Pour contrôler l'apparence de la forme des plis, nous utilisons deux paramètres affectant leur largeur et leur profondeur :
- Le rayon de courbure $R(u)$, défini le long de chaque courbe de pli
- L'offset $\beta(u)$ qui détermine la portion d'arc circulaire formant le pli.

Nous utilisons le rayon de courbure minimale $R_{\min}$ pour tenir compte du type de tissu. Ce rayon est choisi par l'utilisateur et, de par nos expériences, permet un contrôle suffisamment intuitif du comportement des plis.

Loi d'évolution du rayon des plis

Les rayons effectifs des plis sont en fait fonction de cette limite $R_{\min}$ et du taux de compression calculé à partir du *champ de vecteur de plis*. Le rayon des plis étant une fonction décroissante du taux de compression appliqué.

Nous n'avons pas connaissance de loi physique ou procédurale établie concernant cette relation. C'est pourquoi nous avons conduit notre propre expérience sur différents types de tissus en mesurant le rayon de courbure en fonction du taux de compression appliqué. Basés sur ces expériences (voir fig. 3.14), nous définissons la relation suivante :

$$R(u) = \left(\frac{1 - 2/\pi}{v(u)} \right) R_{\min} \, , \tag{3.8}$$

où $v(u) = \|\mathbf{v}(u)\|$ représente le taux de compression, c'est-à-dire la norme du *champ de vecteur de plis* le long de la courbe de pli. Cette équation empirique signifie que le rayon de courbure de la surface dans la direction du pli est interpolé linéairement entre une valeur nulle lorsqu'il n'y a pas de compression, à $1/R_{\min}$ lorsque le pli est donné par un profil semi-circulaire (c'est à dire lorsque $v(u) = 1 - 2/\pi$).

Calcul de la profondeur du pli

Une fois le rayon du pli fixé, la largeur et hauteur de ce pli doivent être choisies tels que l'ajout de celui-ci minimise la compression mesurée sur le maillage à basse résolution.
Nous modélisons ce comportement en utilisant l'offset β tel que le gain en longueur ΔL du à l'ajout du pli compense la compression locale du maillage.

Un pli compressé au maximum sera ainsi modélisé par un arc de cercle dont le centre se positionne sur la surface du vêtement, alors qu'un pli naissant verra son centre de courbure situé profondément sous cette surface.

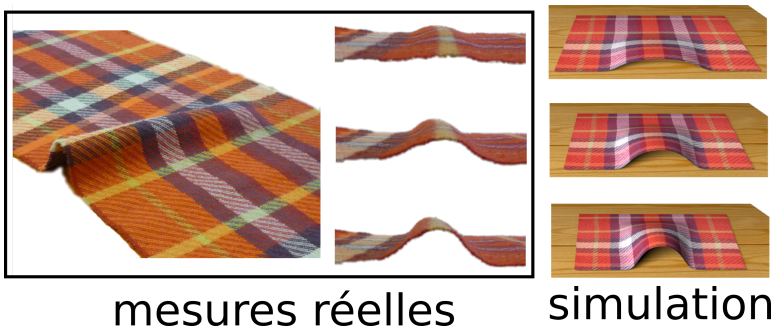


FIGURE 3.14 – *Mesure réelle (gauche) de la courbure en fonction du taux de compression. Ces photos de plis peuvent être comparées aux résultats de notre simulation utilisant l'eq. (3.8).*

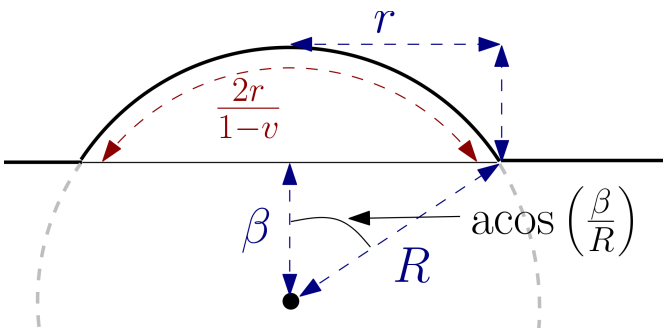


FIGURE 3.15 – *Modélisation d'un pli vue en coupe. La surface de base est schématisée par le plan, alors que le pli consiste en la partie semi-circulaire. Le centre du pli circulaire est situé sous la surface originale à une distance β. Le pli possède un rayon de courbure R.*

Nous montrons en annexe C.1 qu'en supposant que la compression soit constante sur la largeur du pli, la relation liant $\beta(u)$ et le rayon de courbure $R(u)$ pour une compression donnée est la suivante (voir fig. 3.15) :

$$(1 - v(u)) \operatorname{acos}\left(\frac{\beta(u)}{R(u)}\right) - \sqrt{1 - \left(\frac{\beta(u)}{R(u)}\right)} = 0\,. \tag{3.9}$$

Notons que cette relation est valide pour $v(u) \in [0, \frac{2}{\pi}]$. Si $v(u) > 2/\pi$, nous utilisons un profil semi-circulaire en considérant $\beta = 0$, ce qui implique alors une compensation partielle de la compression.

Pour résoudre cette équation en β, nous utilisons la méthode de recherche de racine de Newton [QSS00]. Nous utilisons comme point de départ $\beta = 0$ (squelette sur la surface). Notons qu'il n'existe qu'une unique solution dans l'intervalle $[0, R]$, et que la dérivée de cette expression s'exprime analytiquement. En pratique, la solution est ainsi obtenue rapidement et suffisamment précisément en moins de 5 itérations.

3.3.2 Deformeurs géométriques implicites

Nous souhaitons désormais générer des plis géométriques suivant les courbes tracées dont le profil prend en compte les paramètres décrits précédemment. Nous souhaitons également

gérer de manière automatique les courbes proches l'une de l'autre tout en évitant les auto-intersections et autres artefacts.

Les méthodes procédurales décrites précédemment [CGW+05, DJW+06] utilisent des profils de plis pré-définis ou dessinés manuellement, et ne peuvent donc pas être utilisés pour la génération de plis dans le cas de déformation générale.

L'approche de Popa *et al.* [PZB+09] peut potentiellement gérer plusieurs paramètres de contrôle, cependant la méthode ne permet pas de modéliser aisément les plis spatialement proches ou se mélangeants.

À la place, notre méthode repose sur une nouvelle méthode de déformeurs implicites spécialement définis pour posséder les spécifications souhaitées. La formulation implicite supporte l'interpolation lisse des paramètres des plis, et permet de gérer de manière robuste et sans traitement supplémentaire des interactions entre plis complexes telles que le mélange lisse ou la séparation. De plus, cette formulation évite les collisions entre plis proches en remplaçant de manière implicite l'intersection par le mélange. Les intersections entre les plis et le corps du personnage sont évités pendant la phase de déformation suivant une procédure que nous détaillerons après avoir présenté l'approche générale.

Utilisation de primitives implicites

Rappelons qu'une surface implicite est une isosurface $f(\mathbf{p})$ d'un champ scalaire volumique f, souvent défini comme une fonction décroissante de la distance à un élément source appelé squelette (voir partie 1.1.2).

Pour simplifier la suite de notre exposé, l'iso-valeur utilisée pour définir l'ensemble des surfaces sera prise égale à 1.

Les surfaces implicites sont connues pour leur gestion automatique des mélanges de formes lisses obtenus par simple ajout des champs scalaires générés par différents squelettes spatialement proche.

Nous définissons dans notre cas la géométrie implicite des plis en utilisant les courbes des plis comme squelette.

Il en résulte des cylindres généralisés implicites qui sont alors mélangés ensemble, modélisant ainsi l'appariement et la séparation lisse des plis. La structure obtenue est utilisée en tant que déformeur pour localement modifier la surface initiale autour d'eux en projetant les sommets avoisinants sur les isosurfaces ainsi combinées (voir fig. 3.16).

Primitives de plis

On rappelle (voir partie 1.1.2) que l'utilisation de primitives implicites basées sur des fonctions de plus proche distances à un squelette possède le désavantage de générer des bosses lors des jonctions entre squelettes. Or, on peut noter que les plis de vêtement qui se croisent génèrent un pli final plus large, mais d'amplitude relativement constante.

Afin de modéliser des mélanges de plis exempts des bosses non souhaitées, nous utilisons des surfaces de convolutions. Similairement à l'eq. (1.5), nous définissons une primitive de pli pondérée comme l'intégrale d'une fonction noyaux le long d'une courbe de squelette $c(u)$:

$$f(\mathbf{p}) = \int_u \omega(u)\ \kappa\left(\|c(u) - \mathbf{p}\|\right) \mathrm{d}u\ ,$$

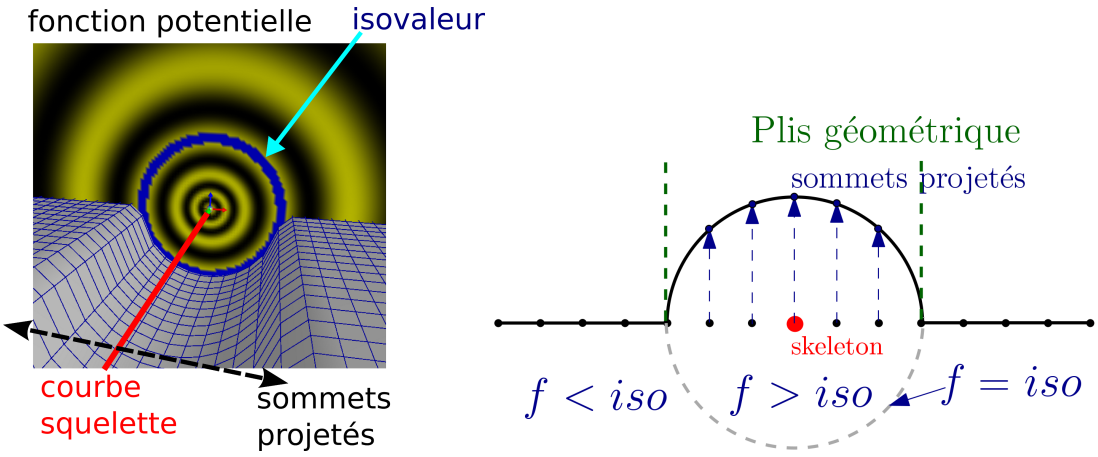


FIGURE 3.16 – *Principe du déformeur implicite projetant les sommets de la surface sous son rayon d'influence sur l'isovaleur de la fonction implicite définie par la courbe.*

où $\omega(u)$ est une fonction de poids contrôlant le rayon du deformeur et κ est une fonction noyaux. Dans notre cas, nous utilisons κ comme le noyau de Cauchy défini en eq. (1.8).

On rapelle que

$$\kappa(x) = \frac{1}{(1+s^2x^2)^2} ,$$

avec, dans le cas présent, $s \simeq 100$ pour des maillages de taille unitaire. Cette valeur permet d'éviter que les plis ne se mélangent à des distances trop éloignées. On rappelle également que l'utilisation du noyau de Cauchy permet de connaitre le résultat exact de l'intégrale sous forme analytique (voir eq. (1.9)).
La pondération par ω permet de contrôler le rayon de la primitive implicite le long du squelette. Notons qu'une définition de ω linéaire par morceaux entre les sommets discrets de la courbe de squelette permet de garder une expression analytique exacte de l'intégrale [ITZkF04].

Pour obtenir le rayon du déformeur souhaité, nous analysons l'expression analytique du noyau de Cauchy intégré sur un segment. En supposant que la courbe de pli soit grande devant la largeur des plis, il est possible d'exprimer analytiquement la relation liant la pondération $\omega(u)$ à donner afin d'obtenir une primitive implicite de rayon $R(u)$ souhaité. Nous montrons en annexe C.2 que cette relation s'exprime par

$$\omega(u) = \frac{2s(1+s^2R(u)^2)^{3/2}}{\pi} . \tag{3.10}$$

Le paramètre d'offset β est utilisé pour positionner le squelette des plis à une distance variable $\beta(u)$ en dessous (ou au-dessus) de la courbe de pli associée afin de permettre la modélisation de plis profonds ou de surfaces.

Déformation de la surface

Nous souhaitons permettre le mélange des plis avec la surface initiale sous-jacente pour laquelle nous ne possédons pas de représentation implicite.
Calculer une telle représentation à partir d'une convolution de surface sur le maillage triangulaire complet est possible, mais très couteux.

À la place, nous introduisons une nouvelle approche permettant le mélange local entre géométries implicites et explicites. Pour cela, nous nous basons sur l'observation que vue d'un point de l'espace suffisamment proche d'une surface lisse, celle-ci peut être approximée par son plan tangent.

Déformeur de maillage

Nous définissons une primitive implicite supplémentaire appelée *déformeur de maillage* dont la contribution à chaque point de l'espace $\mathbf{p}$ est donné par la contribution du plan tangent évaluée au point le plus proche du maillage.

Appelons d la distance entre $\mathbf{p}$ and le point le plus proche du maillage. En appliquant la formulation de l'eq. (1.10) donnant le potentiel d'un plan infini, on a alors

$$f_{\text{mesh}}(\mathbf{p}) = f_{\text{plane}}(\mathbf{p}) = \frac{\pi}{s^2(1+s^2(d(\mathbf{p})+d_0)^2)}\,, \tag{3.11}$$

avec $d_0 = 1/s\sqrt{\pi/s^2-1}$ utilisé pour assurer que l'isosurface associée à l'isovaleur 1 soit positionnée sur le maillage.

La déformation finale consiste à projeter localement les sommets du maillage le long de leur direction normale sur l'isosurface générée par la somme des champs issus de la contribution du *déformeur de maillage* et de celle des primitives de plis.

L'action combiné des plis et du *déformeurs de maillage* est calculée à vitesse interactive et génère des formes de plis plausibles, reflétant la compression appliquée sur le maillage grossier ainsi que des propriétés du tissu.

Pour éviter que les déformeurs de plis ne viennent affecter plusieurs couches du maillage lorsque celui-ci représente une surface plissée, leur contribution est uniquement appliquée sur la portion de surface associée à celle courbe. Au dela de cette distance, leur influence peut être négligée (on prendra en pratique $2R(u)$).

Notons que l'utilisation d'une étape de projection illustrée en fig. 3.16 est nécessaire dans le sens où la sommation du champ de déplacement associé à des courbes de squelettes [SF98] génèrerait des artefacts visibles dans notre cas. À la place, notre schéma prend avantage du comportement exempt de gonflement des surfaces de convolutions lorsque les primitives de plis se touchent.

Subdivision locale du maillage

Pour capturer précisément la géométrie des plis, le maillage est localement subdivisé dans les régions d'influences des courbes de plis. En effet, une subdivision globale génèrerait un nombre important de triangles inutiles rendant l'algorithme plus lent. De plus, l'utilisation d'une subdivision fine, mais non adaptée aux plis risque d'apporter des artefacts visuels lorsque les plis ne sont pas alignés le long des arêtes du maillage [CGW+05, MC10].

La connaissance explicite des courbes de plis nous permet d'en tirer avantage lors de la subdivision en assurant que les arêtes des triangles fins soient alignées le long de ces courbes. Nous décrivons ci-après la méthode de subdivision sur le patron 2D

– Dans une première étape, nous supprimons les triangles du maillage original proche des courbes de plus. Nous considérons pour cela une distance de $4R_{\min}$.

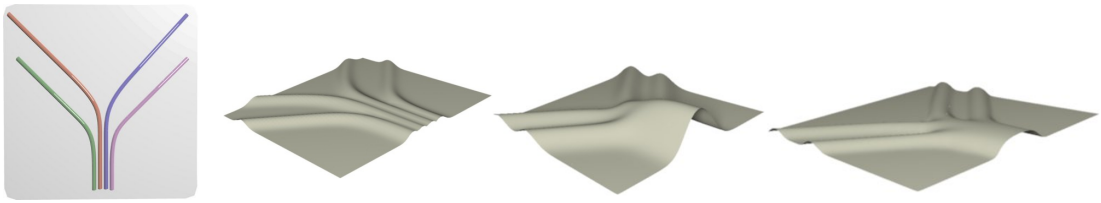


FIGURE 3.17 – *Comportement du déformeur implicite lors du mélange de plis géométriques. De gauche à droite : Surface vue de haut et courbes de plis se rapprochant ; utilisation d'une déformation suivant le maximum de chaque contribution, notez qu'aucun mélange ne se produit jamais ; sommation des déformations associées à chaque courbe de plis : notez la bosse en amplitude lorsque les plis se mélangent ; déformation basée sur une projection des sommets sur la surface implicite définie par le noyau de Cauchy, notez que le pli devient plus large lors du mélange, mais garde une amplitude relativement constante.*

– Dans une seconde étape, nous ajoutons les sommets des courbes de plis dans les régions vides ainsi délimitées. Notons que ces sommets ne sont ajoutés que s'ils sont suffisamment loin l'un de l'autre lorsque des courbes sont proches.
– Dans une troisième étape, nous réalisons une triangulation de Delaunay [She] qui viens remplir les trous réalisés tout en considérant les nouveaux sommets des courbes de plis. Une subdivision automatique des triangles lors de l'algorithme est réalisée afin de satisfaire un critère d'angle.

Des exemples de telles triangulations sont illustrés en fig. 3.18 et 3.19.

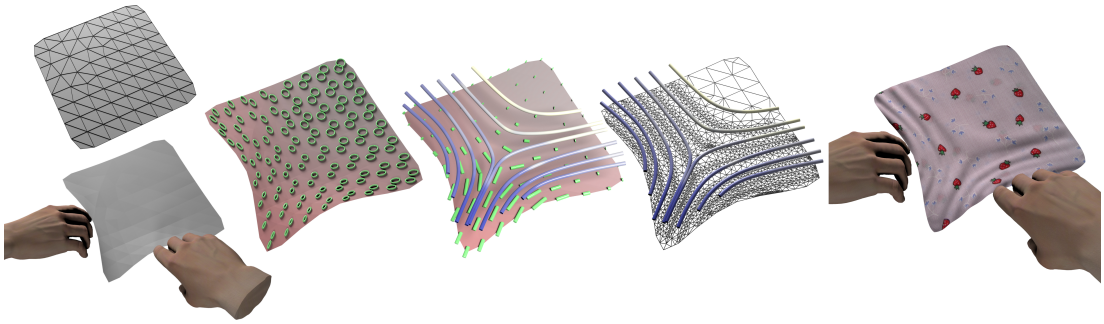


FIGURE 3.18 – *Application de notre méthode de création de plis avec triangulation adaptée. De gauche à droite : Déformation de la surface à basse résolution ; génération des tenseurs de déformations ; création des courbes de plis ; triangulation locale ; génération des plis géométriques.*

Nous pouvons également nous comparer au résultat de Müller et Chentanez [MC10] en fig. 3.20 qui nécessite l'utilisation d'une subdivision globale. Dans notre cas, la connaissance explicite des plis permet d'ajuster le maillage à ceux-ci évitant ainsi les problèmes d'échantillonnage.

Notons que cette méthode est optimale pour le cas statique. Lors d'animations, la modification de la triangulation au cours du temps peut engendrer des défauts visuels. Il est parfois préférable de considérer un maillage constant au cours du temps dans ce cas de figure.

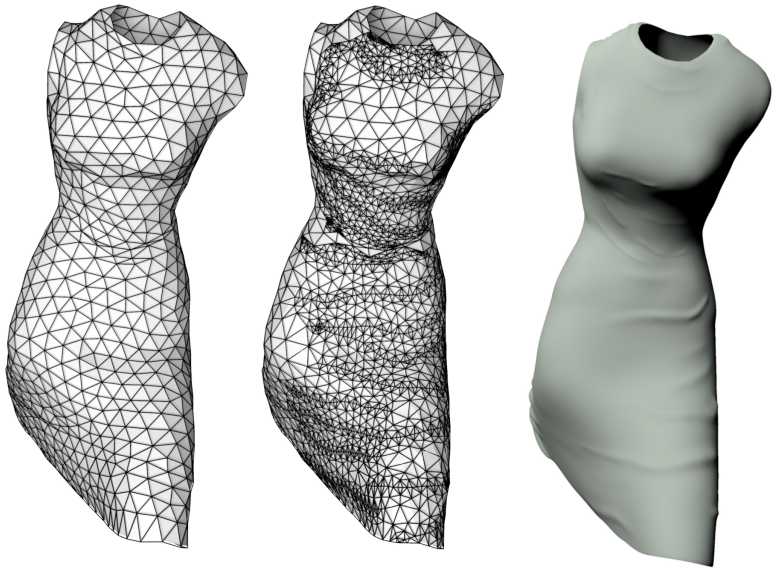


FIGURE 3.19 – *Triangulation locale adaptée sur un cas de robe complète. Gauche : maillage 3D original ; Milieu : Triangulation adaptée aux plis géométriques introduits ; Droite : Apparence finale après application des plis.*

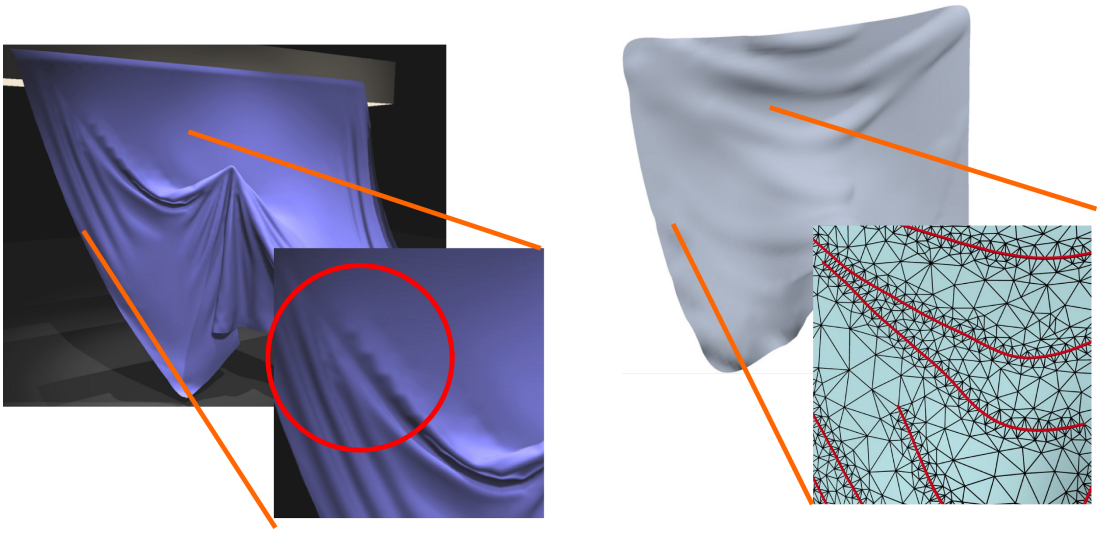


FIGURE 3.20 – *Comparaison de la triangulation avec l'approche de Muller et al. [MC10] et la notre. Dans le cas de [MC10], la subdivision globale peut se révéler peut adaptée aux plis fins et générer des artefacts d'échantillonnages (gauche), alors que notre approche permet d'échantillonner efficacement la surface le long des plis (droite) grâce à la connaissance explicite de la localisation de ceux-ci.*

Direction des plis et évitement des collisions

Nous pouvons noter que les plis peuvent être générés des deux côtés de la surface. En choisissant l'offset approprié et la direction de projection, chaque déformeur de plis peut ainsi être utilisé pour contrôler la direction des plis par rapport à la surface originale.

Dans le cas de vêtement de personnages animés, nous choisissons de considérer la direction externe par défaut pointant vers l'extérieur du personnage. Nous évitons ainsi d'introduire des collisions entre le maillage déformé et le corps du personnage.

Le cas de collisions plus complexes avec présence d'obstacles des deux côtés du maillage grossier peut être traité en projetant les sommets en collisions sur la surface externe de l'objet lors de l'application de la déformation. Nous évitons ainsi les collisions, cependant les plis disparaissent alors, et la compression du maillage grossier n'est plus compensée.

3.4 RESULTATS

Nous appliquons notre algorithme dans divers scénarios, et nous analysons le résultat obtenu en terme d'isométrie, de réalisme visuel, et de temps de calcul.

3.4.1 Mesure d'isométrie

La fig. 3.21 montre le comportement de différents types de tissu simulés en changeant le paramètre de contrôle $R_{\min}$. Nous pouvons voir un morceau de tissu artificiellement compressé unilatéralement à l'avant pour trois différentes valeurs de $R_{\min}$ entrainant différentes formes de plis en terme d'amplitude et de fréquence.

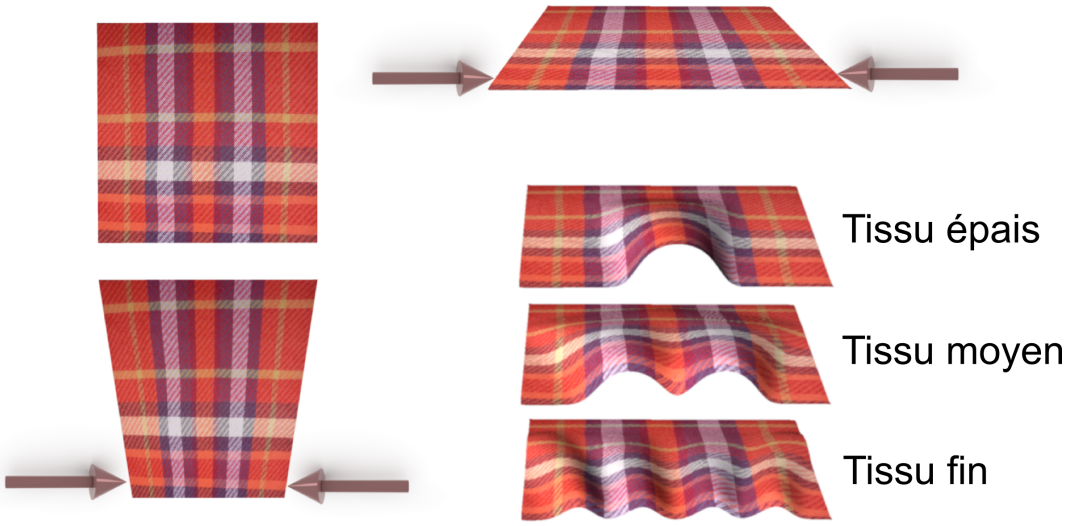


FIGURE 3.21 – *Différents types de déformations d'un tissu suivant la valeur de $R_{\min}$. Gauche : surface originale (haut) et déformée (bas). Droite : Trois types d'effets compensant la même quantité de compression.*

Pour évaluer l'impact de la génération de nos plis sur la correction de compression du tissu, nous mesurons l'élongation avant et après application de notre approche.
Le maillage grossier possède une erreur d'élongation maximale de 36% dans la direction de compression. Notre algorithme permet de restaurer la préservation de longueur avec une erreur de 2% dans cette même direction, le résidu étant dû en grande partie au mélange des plis avec le maillage et entre eux. Dans la direction opposée à la compression, c'est-à-dire le long des plis, l'erreur en longueur reste inférieure à 5%.

Finalement, nous calculons également l'erreur totale d'élongation entre le patron et le maillage obtenu en introduisant la norme L^2 d'erreur sur l'amplitude de compression donnée par v. Nous obtenons dans le cas présent 20% de réduction d'élongation totale par rapport au maillage à basse résolution. Notons que la préservation exacte de l'isométrie n'est pas l'objectif de ce travail.

3.4.2 Réalisme visuel

Nous avons réalisé des comparaisons qualitatives entre nos résultats et des photos réelles pour deux scénarios : Un bout de tissus compressé dans différentes directions en fig. 3.22 ainsi qu'une serviette poussée par la jambe d'un personnage en fig. 3.23.

Comme visualisés pour ces deux exemples, les résultats issus de notre méthode sont comparables visuellement à ceux des photos réelles. La comparaison exacte avec un modèle réel n'est cependant pas possible étant donné que les configurations initiales sont différentes.

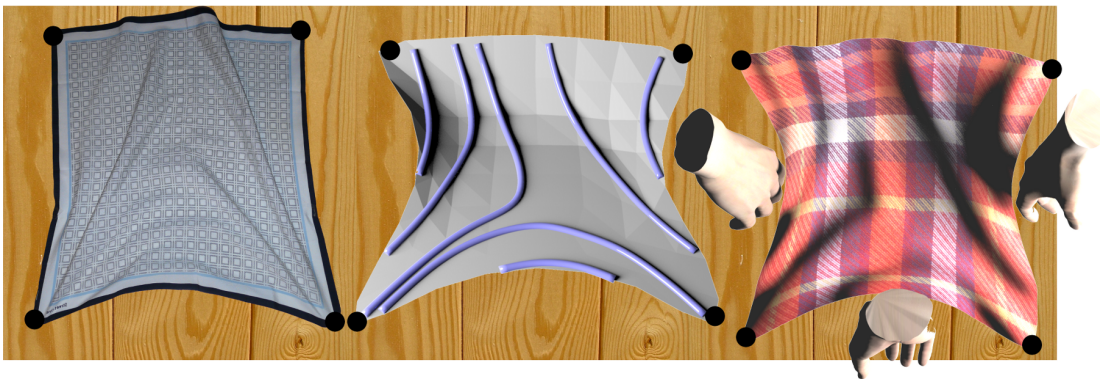


FIGURE 3.22 – *Comparaison visuelle de plis se mélangeants sur un vrai tissu (gauche) et une simulation basée sur notre méthode (droite). La figure du milieu indique la localisation des courbes de plis.*

Il est également possible de comparer nos résultats à des simulations à haute résolution. Nous pouvons ainsi voir en fig. 3.24 que nos résultats permettent de donner un aspect similaire en une fraction du temps nécessaire à la simulation précise. Notons qu'encore une fois, la comparaison exacte n'est pas possible. En effet, la subdivision du modèle à masse ressort modifie le comportement global du tissu le rendant plus flexible. Il est alors nécessaire d'augmenter les constantes de raideurs pour se rapprocher du résultat initial se rapprochant du comportement du modèle à basse résolution par séries d'essais-erreurs. Notons également que le comportement du modèle de collision est différent de par la présence de triangles plus petits.

3.4.3 Application à des vêtements de personnages

Nous avons testé notre approche pour différentes animations de tissus. Les simulations à basses résolutions ont été créées en utilisant le logiciel Blender [Ble]. L'ensemble des modèles correspond à des maillages non parfaitement développables. Nous générons des coutures et paramétrisons ceux-ci également sous Blender.

La figure 3.25 montre des poses d'un T-shirt et d'une robe portée par un personnage animé. Les résultats sont montrés avant, puis après ajout des plis géométriques, et illustrent que cet

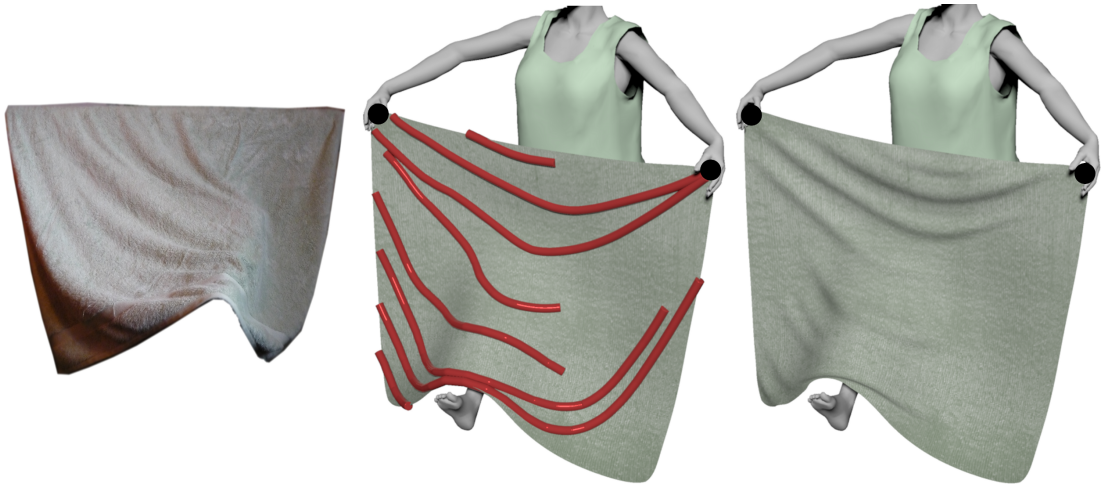


FIGURE 3.23 – *Comparaison entre des plis sur une serviette réelle (gauche), et nos plis créés par notre approche (droite). La figure du milieu indique la localisation des courbes de plis.*

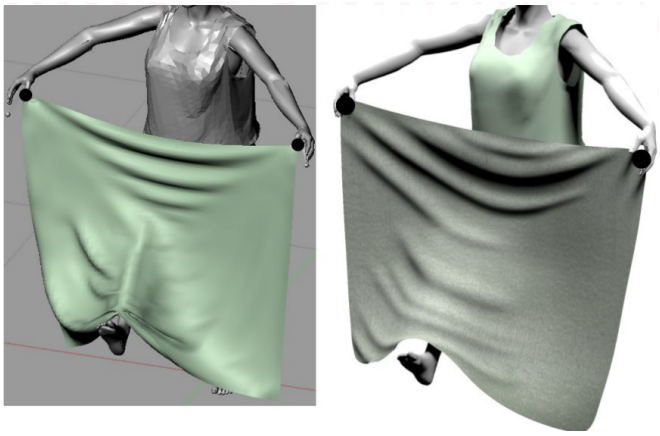


FIGURE 3.24 – *Comparaison entre une simulation à haute résolution (gauche) et nos résultats (droite).*

ajout améliore la plausibilité de la visualisation ainsi que l'impression de dynamique. Notons que la figure de la robe en vert est modélisée par un paramètre de rayon de courbure $R_{\min}$ petit donnant ainsi l'impression d'un tissu très fin présentant de nombreux plis de type soie.

Le rôle du paramètre $R_{\min}$ est également illustré en fig. 3.26 où pour une même compression appliquée sur la simulation d'une serviette, deux types de matériaux peuvent être modélisés.

Notre méthode s'adapte également à l'animation par skinning. Ainsi, nous appliquons une déformation par skinning sur une jambe se pliant comme montré en fig. 3.27. Notre méthode permet d'ajouter des plis aux endroits compressés par le mouvement.

Enfin, la série de figures suivantes représente des poses prises successivement lors d'animations et prouve la versatilité de notre approche pour l'animation de différents vêtements. La fig. 3.28 représente l'animation du T-shirt présentée précédemment. Le T-shirt est volon-

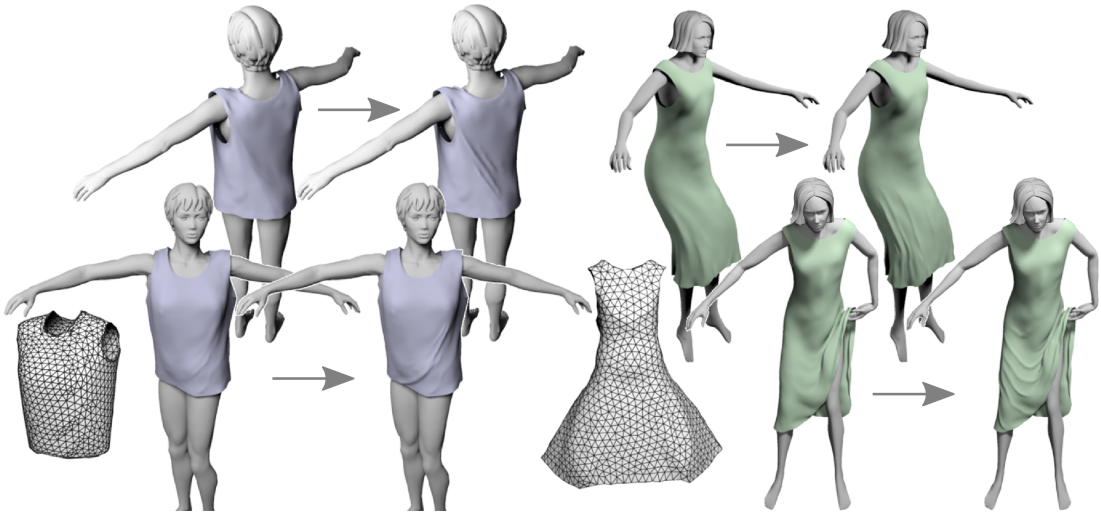


FIGURE 3.25 – *Deux exemples de vêtements portés par un personnage. Dans chaque exemple les poses sont montrées avant ajout des plis géométriques, puis après. Les maillages de référence correspondant à la simulation à basse résolution sont également affichés.*



FIGURE 3.26 – *Deux types de matériaux modélisés en fonction de la valeur de $R_{\min}$. Gauche : Simulation originale avec compression des polygones au niveau des contraintes de la main du personnage. Figures du milieu : Deux cas de types de plis. Le cas où $R_{\min}$ est relativement grand, modélisant ainsi peu de plis larges tels que ceux rencontrés sur une matière de type jeans. La figure suivante représente l'application d'un $R_{\min}$ plus faible. Enfin, la figure de droite est un zoom présentant la géométrie complexe des plis pouvant se mélanger de manière lisse.*

tairement choisi flottant au niveau du bas du ventre afin de permettre à nos plis d'amplifier l'impression de dynamique du mouvement. La fig. 3.29 représente l'ajout de plis sur une animation préexistante appelé *Big Buck Bunny* [Ble] où nous avons ajouté une jupe au caractère animé. La fig. 3.30 représente le personnage dont la robe modélise un effet de soie. On pourra noter la présence de nombreux plis suivant les déformations complexes de cette robe. Enfin, la fig. 3.31 représente la vue arrière d'un personnage, lors d'un mouvement de danse à la dynamique importante et comparée à la simulation à basse résolution. La vue avant est présentée dans l'image de teaser de ce chapitre en p. 141. Notons que l'ensemble des plis ajoutés par notre méthode améliore nettement le dynamisme de la scène.

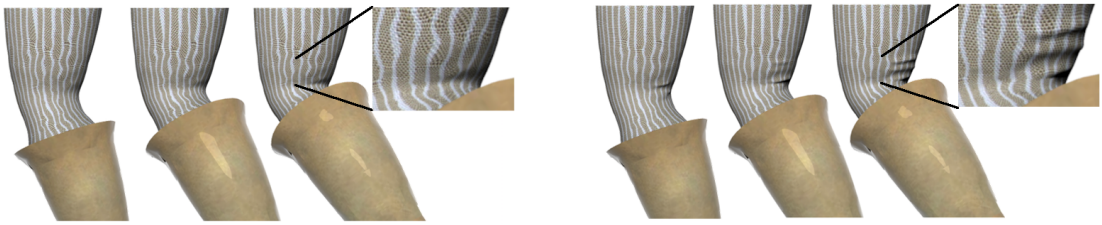


FIGURE 3.27 – *Application de notre génération de plis sur une animation par skinning. La série de figures de gauche représente une jambe se pliant par skinning standard. On pourra noter l'absence totale de plis, et donc la compression des triangles au niveau de l'articulation. La série de figures de droite représente le résultat après ajout de nos plis dans les régions compressées. Les plis s'orientent automatiquement suivant la direction principale de compression et correspondent à une direction plausible de plis au niveau de la jambe.*



FIGURE 3.28 – *Animation d'un T-shirt*

TEMPS DE CALCUL

Les temps de calcul associés aux différents exemples présentés sont rassemblés dans le tableau 3.1.

On pourra noter que notre implémentation actuelle permet d'envisager la mise en place d'une telle méthode dans des outils de modélisation de tissus. Cependant, elle reste encore trop couteuse pour une application directe dans des jeux vidéos. L'étape la plus couteuse reste cependant la projection des sommets sur la surface implicite afin de générer la géométrie des plis. Cette étape est trivialement parallélisable et il est possible de prendre un net avantage d'implémentation future optimisée sur carte graphique afin d'améliorer les performances.
Un premier test d'implémentation en OpenCL a été réalisé et permet la mise en place d'une animation de l'ordre de 5 à 10 fps ce qui laisse envisager des améliorations notables après mise en place d'optimisations spécifiques.

Modèles	fig.	Nbr sommets originaux	tps calcul courbes (ms)	Nbr sommets projetés	tps projection (ms)
Serviette	(3.23,3.24)	80	50	300	75
T-shirt	(3.25,3.28)	1168	800	1500	750
Pantalon	(3.27)	1112	400	500	400
Jupe lapin	(3.29)	364	300	1000	700
Robe danseuse	(3.31, p.141)	931	700	900	600
Robe soie	(3.25,3.30)	1249	850	3500	2700

TABLE 3.1 – *Temps de calculs de la génération des plis.*

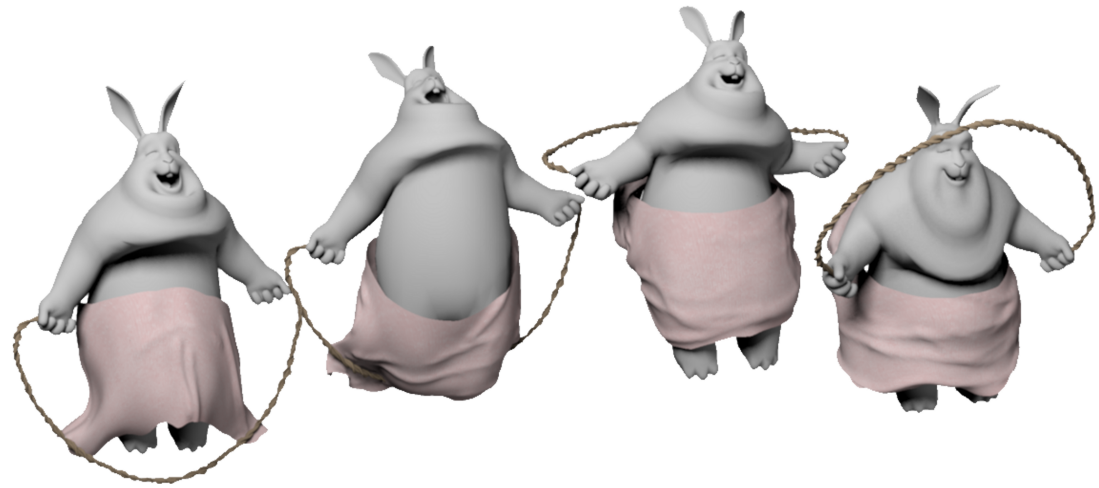


FIGURE 3.29 – *Ajout d'une jupe sur une animation pré-existante [Ble].*



FIGURE 3.30 – *Animation d'une matière de type soie lors d'un mouvement complexe.*



FIGURE 3.31 – *Animation dynamique d'une robe vue de dos. La ligne du haut représente la simulation à basse résolution, la ligne du bas représente notre résultat après ajout de nos plis.*

BILAN ET TRAVAUX FUTURS SUR L'AJOUT DE PLIS DE VÊTEMENTS

Nous avons présenté une méthode de génération de plis animés s'ajoutant en tant que post-calcul sur une animation à basse résolution telle qu'un modèle physique simplifié ou un skinning.
Notre approche permet de garder l'apparence globale de l'animation originale tout en améliorant de manière automatique sa plausibilité par l'ajout de détails géométriques.
Les plis ajoutés permettent de limiter les compressions irréalistes générées par le modèle de déformation précédant tout en accentuant l'impression de dynamique.
La séparation entre déformation physique globale du modèle, et détails géométriques des plis permet de contrôler l'apparence ce ceux-ci de manière intuitive. Différents types de matériaux peuvent ainsi être modélisés suivant l'épaisseur des plis tout en gardant une apparence globale identique.

La méthode proposée est plus rapide qu'une simulation à haute résolution et permet sa mise en application dans des logiciels de modélisation. L'implémentation L'approche proposée ici se voulait la plus automatique possible. Une future piste d'amélioration possible consiste à incorporer davantage de contrôle artistique concernant la forme des plis au cours du temps, ainsi que leur position et leur amplitude qui peuvent dans certains cas ne pas se conforter strictement aux informations de déformations du maillage.

Une autre piste de recherche concerne la prise en compte des élongations. En effet, seule la demie partie < 1 des valeurs propres des tenseurs ont été utilisées pour limiter les élongations. Il serait aisé de considérer également la part > 1 caractéristique d'un étirement. Nous pourrions ainsi prendre en compte les plis de tension liés à l'élongation que l'on peut également observer sur les textiles.

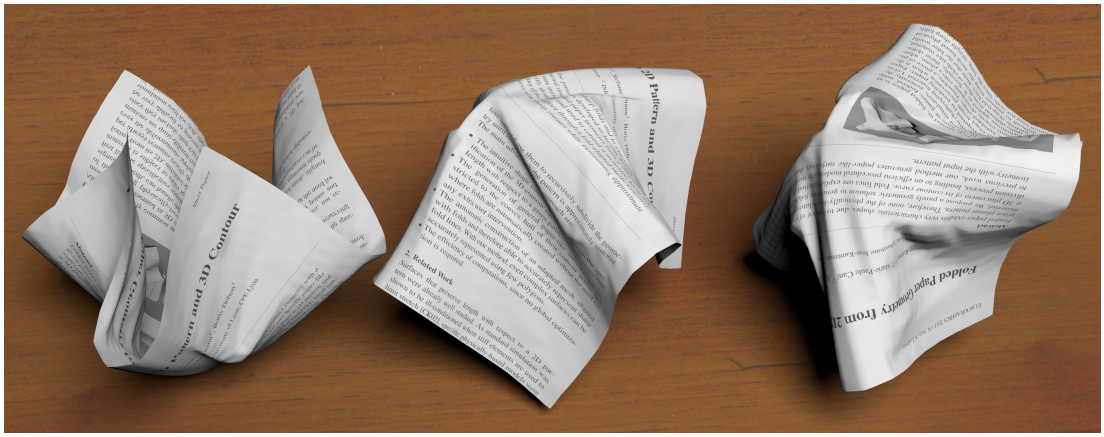
À l'heure actuelle, nous ne pouvons considérer que des plis formés par des profils semi-circulaires ce qui limite la compression maximale de longueur pouvant être corrigée. Dans le cas de compression large, il serait avantageux de pouvoir modéliser des profils de plis plus complexes multivalués dans la direction normale à la surface. L'utilisation de déformeurs implicites à plusieurs squelettes ou à champ de potentiel plus complexe pourrait également être une piste d'avancée future.

Finalement, notre approche ne permet pas de converger vers une préservation d'isométrie stricte. En effet, la perte de longueur n'est que localement compensée dans la direction orthogonale au pli. Certaines matières très rigides nécessitent cependant la préservation stricte des contraintes de longueur. Un cas de tel matériau est par exemple celui du papier qui, de par sa préservation stricte de la contrainte de longueur va venir se froisser plutôt que de créer des plis lisses. Le chapitre suivant s'intéresse à la modélisation rapide de telles surfaces en considérant cette fois la préservation des longueurs dans toutes les directions.

CHAPITRE

4

MODELISATION DE SURFACES ISOMETRIQUES A UN PATRON : APPLICATION AU PAPIER FROISSÉ.



LES matériaux tels que le papier, le métal, le cuir, etc, présentent des contraintes très fortes d'inextensibilité. Ils ne peuvent ni se compresser, ni s'étirer dans aucune direction. Ces matériaux possèdent ainsi la propriété de pouvoir se déplier à plat sur leur patron d'origine sans aucun changement de longueur ; les surfaces associées sont qualifiées de *développables*.

Bien que le papier soit un matériau commun dans la vie réelle, il n'est que peu représenté dans le domaine virtuel tel que dans les films d'animation ou les jeux vidéos. En effet, l'absence d'outils de modélisation spécifiques à de tels matériaux rend leur génération fastidieuse pour les artistes numériques. La structure microscopique particulière du papier génère un comportement macroscopique de la surface dit *plastique* reconnaissable visuellement à la présence de plis francs dues à la rupture de fibres, et liés à l'impossibilité d'étirer ou de compresser la surface.

La modélisation virtuelle de telles surfaces est complexe et généralement lourde en temps de calcul du fait de la nature intrinsèquement globale et non linéaire du respect d'une contrainte d'isométrie. De ce fait, les approches existantes sont le fruit de compromis entre précision du respect de l'isométrie et temps de calcul. De plus, les matériaux totalement inextensibles tels que le papier présentent des lignes de plis francs. Modéliser ce type de surface sans un maillage adapté le long de ces plis résulte en une perte de temps de calcul dans les régions planes, et en une mauvaise approximation le long des discontinuités du plan tangent.

L'originalité de l'approche présentée dans ce chapitre consiste à modéliser ces matériaux inextensibles en subdivisant une surface récursivement le long de ses lignes de plis. La sélection des arêtes à ajouter successivement est réalisée automatiquement grâce à un critère d'étirement local, et à chaque étape, la nouvelle géométrie générée améliore l'isométrie. Lorsque cette approche locale ne permet plus d'améliorer le critère d'isométrie, nous entrelaçons entre les étapes de subdivision des phases de relaxation globale des sommets actuels, à l'aide d'une déformation limitant les compressions.

Les avantages de notre méthode sont :
- Une construction de surface quasi isométrique au patron, qui approxime la courbe de bord 3D donnée en entrée et l'améliore si elle n'était pas parfaitement isométrique au bord du patron.
- La génération d'une géométrie développable à partir de courbes de bords, non restreinte, contrairement aux méthodes existantes, au polygone convexe s'appuyant sur la frontière de la surface.
- Une génération de plis de qualité, grâce à un maillage construit automatiquement le long des arêtes vives au cours de la subdivision. Cela évite tout problème de résolution et le nombre de triangles du maillage final reste faible.
- Un temps de calcul raisonnable ($\simeq$ 1s) pour modéliser une surface plissée grâce à une approche récursive de subdivision en sous-problèmes locaux tant que possible, entrelacée de relaxations linéaires globales si nécessaire.
- Une approche purement géométrique pouvant traiter des surfaces non lisses comme le papier, qui serait très difficile à obtenir par simulation physique.

Contrairement aux approches existantes présentées en partie 1.4.3, notre méthode permet de prendre en compte à la fois une courbe de bord 3D et un patron 2D, avec lequel nous préservons l'isométrie. Les surfaces engendrées peuvent sortir de l'enveloppe convexe de leur bord. De plus, un maillage adapté est engendré par découpage récursif le long des lignes où les plis

sont les plus marqués, ce qui rend la méthode applicable à la modélisation de papier froissé.

Dans un premier temps, nous présenterons une méthode de génération interactive de bords 3D préservant les propriétés de développabilité vis-à-vis d'un patron 2D.
Ce bord 3D est ensuite utilisé en tant que donnée d'entrée dans un algorithme de subdivision de type *divide and conquer* convergent vers une surface de plus en plus isométrique au patron originel. L'algorithme génère des lignes de plis francs de manière automatique lorsque les distances sont compressées. La troisième partie présente différents résultats obtenus sur des surfaces simples et froissées afin de valider quantitativement l'apparence visuelle de surfaces de papier. La préservation d'isométrie est quantifiée, et la robustesse de la méthode est montrée sur des extensions dans le cas de bords 3D non compatibles avec la génération d'une surface isométrique. Enfin, les différentes limitations et extensions possibles sont discutées dans la dernière partie.

Les contributions principales de ce chapitre ont été présentées dans les publications suivantes : [RCHT10], [RCHT11].

4.1 GÉNÉRATION D'UN BORD 3D CONTRÔLABLE

Déformation d'un bord

Nous souhaitons tout d'abord être en mesure de générer de manière guidée une courbe 3D (voir illustration en fig. 4.1) sur laquelle va s'appuyer une surface développable que nous allons générer. Cette surface devra être isométrique au patron 2D connu initialement.

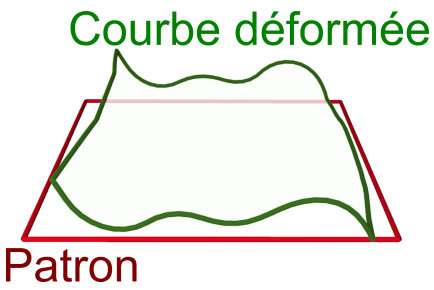


FIGURE 4.1 – *Déformation d'une courbe 3D (vert) sur laquelle s'appuie une surface développable isométrique au patron (rouge).*

Pour cela, nous proposons de déformer la courbe de bord de ce patron, ou d'une surface isométrique à celle-ci, à l'aide d'une déformation adéquate h. Cette déformation h doit en particulier déformer le bord original en une courbe 3D cohérente vis-à-vis de son interpolation par une surface isométrique.

Il reste alors à définir quels sont les critères à respecter pour qu'une courbe 3D soit le bord d'une surface isométrique à un patron planaire donné. Nous noterons que la littérature sur ce type de déformation, potentiellement non dérivable, est peu développée. Nous pouvons cependant définir un ensemble de critères nécessaires pour l'obtention d'une déformation h admissible.

Critères nécessaires

Soit Γ une courbe fermée de l'espace modélisant le bord de la surface 3D souhaitée. Cette courbe est paramétrée par u que l'on considérera arbitrairement compris dans $[0, 1]$, avec $\Gamma(0) = \Gamma(1) \in \mathbb{R}^3$. Soit $\overline{\Gamma}$ la courbe correspondant au bord du patron 2D.

On cherche la déformation h telle que $\Gamma = (h \circ \overline{\Gamma})$. Pour cela, les critères suivants sont nécessaires

– h préserve localement (et donc globalement) les longueurs du bord.

$$\forall u \in [0,1]\ ,\ \ (\Gamma, \overline{\Gamma}) \text{ dérivables } \Rightarrow \|\Gamma'(u)\| = \|\overline{\Gamma}'(u)\|\ . \tag{4.1}$$

– h n'introduit aucune élongation de distance entre deux points du bord.

$$\forall (u_1, u_2) \in [0,1]^2\ ,\ \ \|\Gamma(u_1) - \Gamma(u_2)\| \leq \|\overline{\Gamma}(u_1) - \overline{\Gamma}(u_2)\|\ . \tag{4.2}$$

– h ne génère pas d'extension angulaire aux points de non-dérivabilitées (sommets anguleux)

$$\forall u \in [0,1]\ ,\ (\Gamma, \overline{\Gamma}) \text{ anguleux } \Rightarrow \text{angle}\left(\Gamma'_{\mathrm{d}}(u), \Gamma'_{\mathrm{g}}(u)\right) \leq \text{angle}\left(\overline{\Gamma}'_{\mathrm{d}}(u), \overline{\Gamma}'_{\mathrm{g}}(u)\right)\ , \tag{4.3}$$

où Γ'_{d} représente la dérivée à droite (resp. Γ'_{g} à gauche).

Paramétrisation par abscisse curviligne

Tout d'abord, étant donné 2 courbes Γ et $\overline{\Gamma}$, il est nécessaire de connaitre leur correspondance afin d'effectuer les mesures de distances présentées précédemment.

En supposant que l'on connaisse cette correspondance en un point, nous souhaitons construire une paramétrisation cohérente. Pour cela nous considérons la paramétrisation induite par l'abscisse curviligne relative. Cette approche possède les avantages suivants :

– Rapide à calculer sur une courbe polygonale.
– Paramétrage cohérent supportant le cas de courbes non parfaitement isométrique.

Appelons $t \in [t_0, t_1]$ (resp. $\bar{t} \in [\overline{t_0}, \overline{t_1}]$) un paramétrage quelconque de Γ (resp. $\overline{\Gamma}$). On définit alors la nouvelle paramétrisation par le changement de variable

$$u(t) = \frac{\int_{t_0}^{t'} \|\Gamma'(t')\|\, \mathrm{d}t'}{\int_{t_0}^{t_1} \|\Gamma'(t')\|\, \mathrm{d}t'}\ .$$

Calculs des critères de dévelopabilité

En pratique, nous utilisons des courbes polygonales définies par leurs N_s sommets discrets. La condition de préservation locale de longueur peut être exprimée discrètement pour chaque segment :

– Le critère de préservation de longueur du bord de l'eq. (4.1) correspond à une évaluation de distance sur chaque segment de la courbe polygonale.
– Le critère de non-élongation de distance de l'eq. (4.2) correspond à une évaluation de distance pour chaque paire de sommets de la courbe.
– Le critère de non-extension angulaire de l'eq. (4.3) correspond à une évaluation d'angle en chaque sommet de la courbe.

Application à la déformation de courbes

Étant donné une courbe polygonale, nous mettons en place une méthode de déformation interactive de celle-ci tentant de préserver les critères de cohérences définis précédemment.

Considérons une courbe polygonale fermée de N_s sommets de positions $(\mathbf{p}_i)^0_{i\in[\![1,N_s]\!]}$. Nous déformons cette courbe en déplacant directement un sous-ensemble de sommets indicés par $\mathcal{J} = (j_1, j_2, \ldots, j_m)$. Notons $\mathbf{p}^f_{j_k}$ la position finale souhaitée du $j_k$ième sommet.

Nous cherchons alors une nouvelle courbe formée des sommets $(\mathbf{p}^1_i)_{i\in[\![1,N_s]\!]}$ tel que ceux-ci satisfassent aux contraintes, c'est à dire que

$$\forall j \in \mathcal{J}\ ,\quad \|\mathbf{p}^1_j - \mathbf{p}^f_j\| \text{ soit minimale,}$$

et tel que la courbe puisse préserver les critères de développabilitées de la surface s'y appuyant.

Ces critères s'exprimant de manière non linéaire, nous proposons une approche itérative par projection sur la contrainte. Pour cela, nous considérons la nouvelle courbe comme formée par :

$$\begin{cases} \forall j \in \mathcal{J}\ ,\quad \mathbf{p}_j = \mathbf{p}^f_j \\ \\ \forall j \notin \mathcal{J}\ ,\quad \mathbf{p}_j = \mathbf{p}^0_j \end{cases}$$

Puis nous projetons les sommets sur les contraintes issues des eq. (4.1), (4.2) et (4.3). Pour différencier les positions avant et après projection sur la contrainte, nous notons par $\mathbf{p}'$ les sommets déformés.

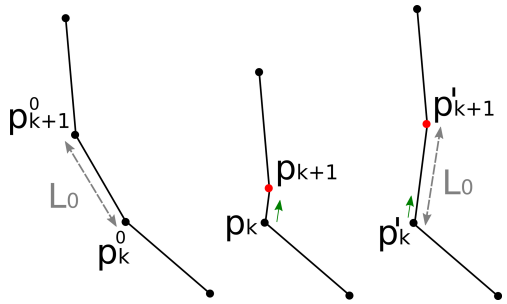


FIGURE 4.2 – *Preservation de longueur d'une courbe polygonale en forçant les segments à garder leur longueur initiale le long de leur direction.*

Pour cela, on considère chaque segment du polygone, et l'on contraint ceux-ci à respecter leur longueur initiale comme illustré en fig. 4.2.

$$\begin{cases} & \mathbf{p}'_1 \quad = \mathbf{p}_1 \\ \forall k \in [\![2, N_s]\!]\ , & \mathbf{p}'_{k+1} \quad = \mathbf{p}'_k + \|\mathbf{p}^0_{k+1} - \mathbf{p}^0_k\| \frac{\mathbf{p}_{k+1} - \mathbf{p}_k}{\|\mathbf{p}_{k+1} - \mathbf{p}_k\|}, \end{cases}$$

Comme ce parcours suivant les indices k croissants n'assure pas de retomber sur une courbe fermée, nous appliquons également cet algorithme dans le sens inverse de parcours des indices k, et moyennons le résultat des deux courbes trouvées tout en assurant que la courbe résultante reste fermée. On notera que la préservation de longueur n'est alors pas exactement respectée.

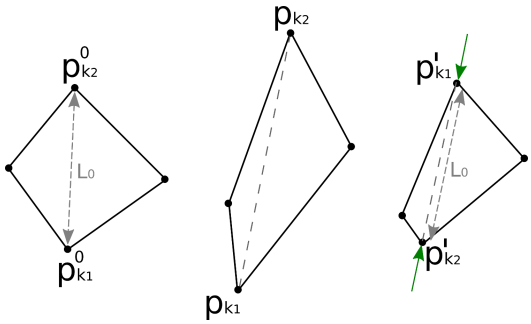


FIGURE 4.3 – *Non élongation d'une courbe en forçant les paires de sommets à ne pas dépasser leur distance originale.*

Dans un second temps, chaque paire de sommets est contrainte à ne pas générer d'extension de distance. Dans le cas d'une détection d'extension, les deux sommets

en cause sont alors rapprochés mutuellement en direction de leur barycentre (voir fig. 4.3). C'est à dire

$$\|\mathbf{p}_{k_1} - \mathbf{p}_{k_2}\| > \|\mathbf{p}^0_{k_1} - \mathbf{p}^0_{k_2}\| \Rightarrow \mathbf{p}'_{k_{1/2}} = \frac{\mathbf{b}-\mathbf{p}_{k_{1/2}}}{\|\mathbf{b}-\mathbf{p}_{k_{1/2}}\|}\frac{\Delta L}{2} ,$$

$$\begin{cases} \Delta L & = \|\mathbf{p}_{k_1} - \mathbf{p}_{k_2}\| - \|\mathbf{p}^0_{k_1} - \mathbf{p}^0_{k_2}\| \\ \mathbf{b} & = (\mathbf{p}_{k_1} + \mathbf{p}_{k_2})/2 . \end{cases}$$

Enfin, similairement, si une extension angulaire entre deux segments est détectée, alors ceux-ci subissent une rotation telle que le surplus angulaire soit annulé comme illustrée en fig. 4.4.

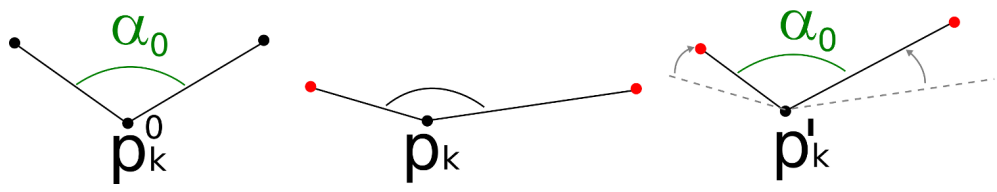


FIGURE 4.4 – *Non élongation angulaire d'une courbe.*

Un exemple d'utilisation de cet algorithme pour la déformation d'une courbe à partir de son patron est illustré en fig. 4.5.

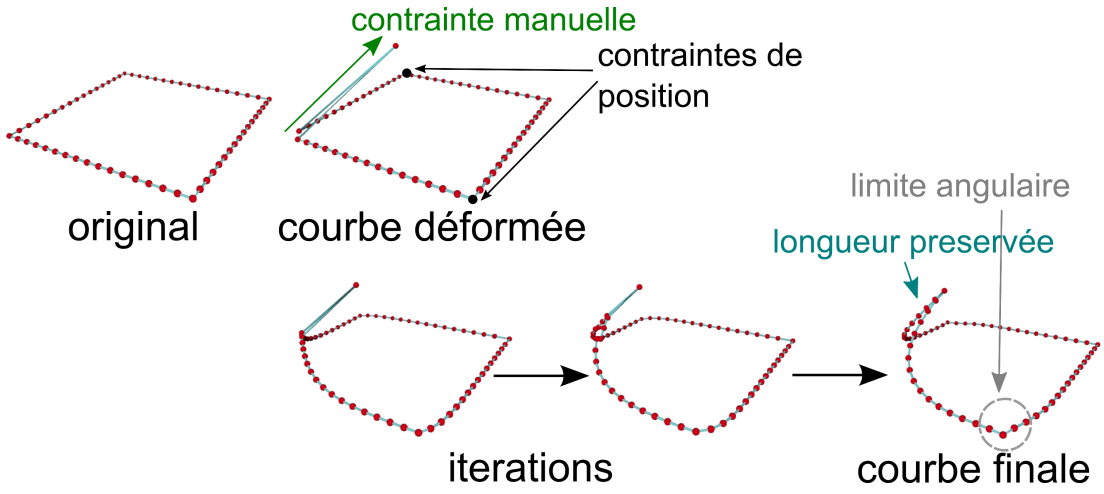


FIGURE 4.5 – *Déformation isométrique d'une courbe à partir de son patron. La courbe déformée ne possède pas d'extension de longueur ou d'angle. Elle fournit ainsi un support cohérent pour la construction d'une surface isométrique au patron original.*

Limitation de cette approche de déformation

On notera que cette approche permet de modéliser des déformations simples d'une courbe en préservant les propriétés de developpabilités surfaciques. Cependant, le contrôle d'une telle courbe n'est pas évident lorsque le nombre de contraintes augmente. Dans le cas de modélisation de bords complexes, cette approche reste relativement limitée et n'offre ni certitudes de

convergence, ni de conditions suffisantes de la développabilité. D'autres pistes de déformation de contours de surfaces développables restent donc à l'étude.

Pour réaliser des bords possédant de nombreuses contraintes, nous proposons de considérer le bord d'une surface développable issue d'une simulation numérique. Le principe est le suivant :

1. Le patron est triangulé en une surface suffisamment détaillée
2. Cette surface est intégrée dans une simulation numérique de tissu afin de la déformer.
   – La simulation est de type masse-ressort implémentée par le logiciel Blender [Ble].
   – Les ressorts sont choisis très raides afin de n'introduire qu'un minimum d'élongation.
3. Une fois la simulation terminée, le bord de la surface déformée est extrait et peut servir de point d'entrée à notre modélisation. On notera qu'il est possible de mixer la déformation par simulation numérique pour la déformation principale avec notre déformation de courbe géométrique pour ajuster certaines parties du bord localement.

Nous avons donc présenté une méthode de génération et de déformation de bord 3D interpolant une surface isométrique à un patron donné. Le bord 3D obtenu est isométrique et ne possède pas d'élongation par rapport à son patron. De plus, sa paramétrisation est identique au bord original ce qui permet d'établir une correspondance entre les deux courbes. La suite de ce chapitre présente notre algorithme de génération de la surface développable interpolant ce bord. Dans un premier temps, nous présentons l'approche de subdivision récursive.

4.2 SUBDIVISION RÉCURSIVE D'UNE SURFACE DE PAPIER FROISSÉ

Nous rappelons que nous considérons comme entrée un patron formé par un polygone convexe du plan noté $\overline{\Gamma}$ paramétrée par son abscisse curviligne relative $u \in [0,1]$ ainsi qu'une courbe 3D Γ représentant le bord d'une surface isométrique au patron. La sortie souhaitée consiste en une surface interpolant ce bord Γ quasi-isométrique au patron et ressemblant visuellement à une surface en papier froissé.

4.2.1 Génération de surface isométrique par diviser pour régner

L'agorithme que nous proposons est inspiré d'une approche de type *divide-and-conquer*. Pour cela, nous insérons itérativement des lignes droites ou des courbes entre deux sommets sélectionnés du contour (voir fig. 4.6). Le choix de ces points particuliers du contour ainsi que le profil de la courbe ajoutée entre eux sont basés sur des critères de préservation de longueurs. Nous proposons une approche de subdivision récursive qui résulte en une succession de surfaces de plus en plus isométrique au patron original.
Lorsque l'insertion de courbes de séparation est terminée, la surface finale est générée par triangulation de l'ensemble des régions formées par les étapes de subdivision.

Le principe général de l'approche par subdivision est le suivant :

1. Trouver le segment $[\overline{\Gamma}(u_1), \overline{\Gamma}(u_2)]$ du patron devant être traité en priorité. Son image 3D est alors décrite soit par un segment (fig. 4.6 milieu), soit par une courbe 3D.
2. Subdiviser le domaine du patron défini par $\overline{\Gamma}$ en deux composants $\overline{\Gamma}^1$ et $\overline{\Gamma}^2$. Les contours 3D correspondants Γ^1 et Γ^2 leurs sont respectivement associés (fig. 4.6 gauche et droite).

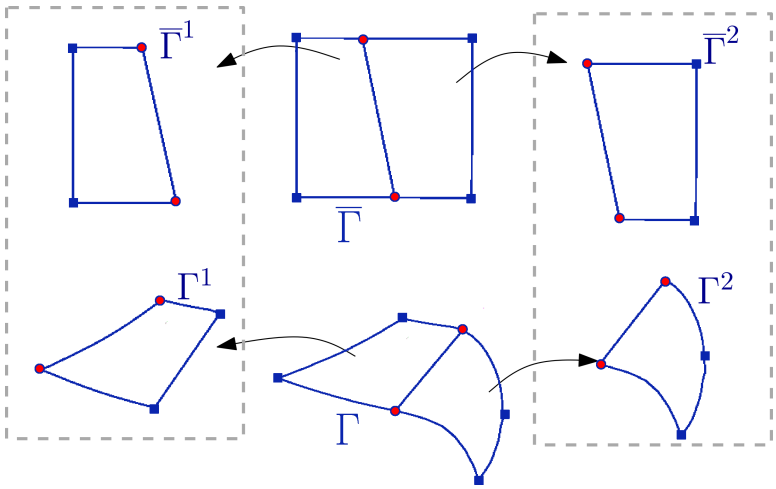


FIGURE 4.6 – *Étapes de subdivision recursives : le patron 2D et le contour 3D correspondant (milieu) sont subdivisés en deux parties (gauche et droite) le long d'une ligne de pli qui est représentée par un segment dans l'espace du patron.*

3. L'opération de subdivision est répétée pour les couples patron+bord3D : $(\overline{\Gamma}^1, \Gamma^1)$ et $(\overline{\Gamma}^2, \Gamma^2)$ considérés comme entrée de l'algorithme.

Des lignes droites sont introduites en priorité dans le cas où le patron et sa correspondance 3D possèdent la même distance euclidienne entre deux positions. Lorsqu'une telle correspondance n'existe pas et que la distance 3D est plus faible que celle du patron, nous sommes alors en présence d'une surface plissée. Dans ce cas, nous permettons à la surface de sortir de l'enveloppe convexe définie par ses contours afin de restaurer l'iométrie au patron original. Pour ce type de modèle, une courbe de plis est alors insérée. Le profil de cette courbe est déterminé par une minimisation améliorant la préservation d'isométrie au fil des étapes de subdivision. L'ensemble des courbes de plis (lignes droites et courbes) est constitué par des segments polygonaux dont les sommets définissent le maillage final. De ce fait, il n'est pas nécessaire d'introduire d'étape de remaillage, car les arêtes du maillage sont alignées le long de ces courbes de plis. En fonction de la géométrie locale du maillage, des arêtes vives peuvent être introduites sans difficulté. L'illustration des étapes de construction d'une telle surface est présentée en fig. 4.7.

4.2.2 Calculs des lignes de plis

Cas de lignes droites en 3D

Nous pouvons tout d'abord noter que le segment $[\Gamma(u_1), \Gamma(u_2)]$ est entièrement contenu dans la surface [1] si et seulement si

$$\|\Gamma(u_1) - \Gamma(u_2)\| = \|\overline{\Gamma}(u_1) - \overline{\Gamma}(u_2)\| \,. \tag{4.4}$$

L'impliquation est triviale. En effet, les deux segments $[\overline{\Gamma}(u_1), \overline{\Gamma}(u_2)]$ et $[\Gamma(u_1), \Gamma(u_2)]$ ont la même longueur par définition de la préservation d'isométrie.

1. On rappelle ici que le bord 2D est supposé convexe.

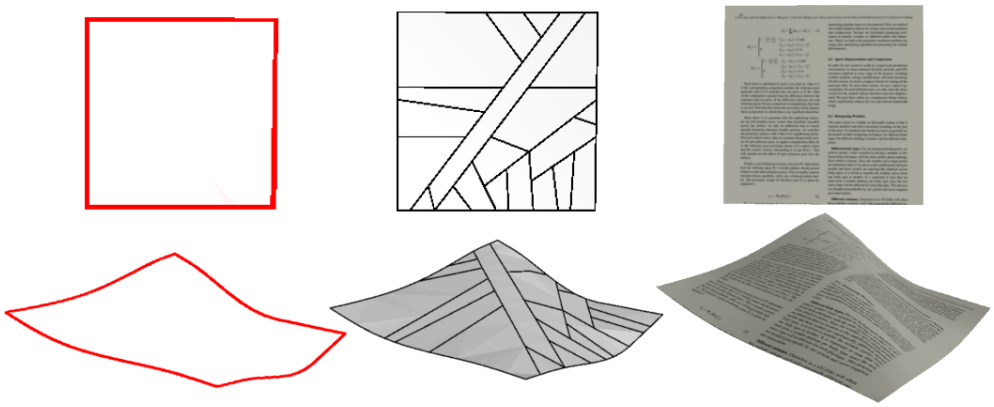


FIGURE 4.7 – *Construction d'une surface de papier. De gauche à droite : contour initiaux (patron 2D et courbe 3D), insertion récursive de lignes de plis, surface résultante.*

La réciproque est également vraie car la seule courbe reliant deux positions $\Gamma(u_1)$ et $\Gamma(u_2)$ et dont la longueur est égale à $\|\Gamma(u_1) - \Gamma(u_2)\|$ est forcément ce segment de droite.

En pratique, nous souhaitons pouvoir traiter des courbes respectant les critères de developpabilités de manière approximative. L'égalité stricte de l'eq. (4.4) n'est donc pas cherchée exactement. Nous définissons ainsi la fonction d'élongation relative

$$\mathcal{R}_{\overline{\Gamma},\Gamma} : \left\{ \begin{array}{ccc} [0,1]^2 & \rightarrow & \mathbb{R} \\ (u_1,u_2) & \mapsto & \left(\frac{\|\Gamma(u_1)-\Gamma(u_2)\|}{\|\overline{\Gamma}(u_1)-\overline{\Gamma}(u_2)\|} - 1 \right) \end{array} \right. .$$

Une recherche robuste d'une paire de sommets non compressée peut se réaliser en définissant

$$(u_1^m, u_2^m) = \text{argmax}_{(u_1,u_2)\in[0,1]^2} \mathcal{R}_{\overline{\Gamma},\Gamma}(u_1,u_2) \, . \tag{4.5}$$

Si $\mathcal{R}_{\overline{\Gamma},\Gamma}(u_1^m, u_2^m) \geq 0$, alors la paire de points est une règle de la surface, ou a été étendue. La *meilleure* solution pour relier ces deux positions est alors la ligne droite définie par le segment $[\Gamma(u_1^m), \Gamma(u_2^m)]$.

En pratique, étant donné $\overline{\Gamma}$ et Γ, nous cherchons le couple (u_1, u_2) satisfaisant la relation donnée en eq. (4.5). Les courbes étant des polygones à N_s sommets, cela revient à une recherche sur l'ensemble des paires de points. Soit au pire $N_s \times (N_s - 1)/2$ calculs de distance. Un exemple de tracé de segments tendus suivant cet algorithme est illustré en fig. 4.8.

Dans le cas où la condition $\mathcal{R}_{\overline{\Gamma},\Gamma}(u_1^m, u_2^m) \geq 0$ n'est jamais respectée, c'est à dire lorsque

$$\forall (u_1,u_2) \in [0,1]^2 \, , \quad \|\Gamma(u_1) - \Gamma(u_2)\| - \|\overline{\Gamma}(u_1) - \overline{\Gamma}(u_2)\| \geq 0 \, ,$$

cela signifie qu'aucune compression n'a lieu dans cette région. La connection entre ces les deux positions de moindre compression définies par (u_1^m, u_2^m) est alors donnée par une courbe de l'espace.

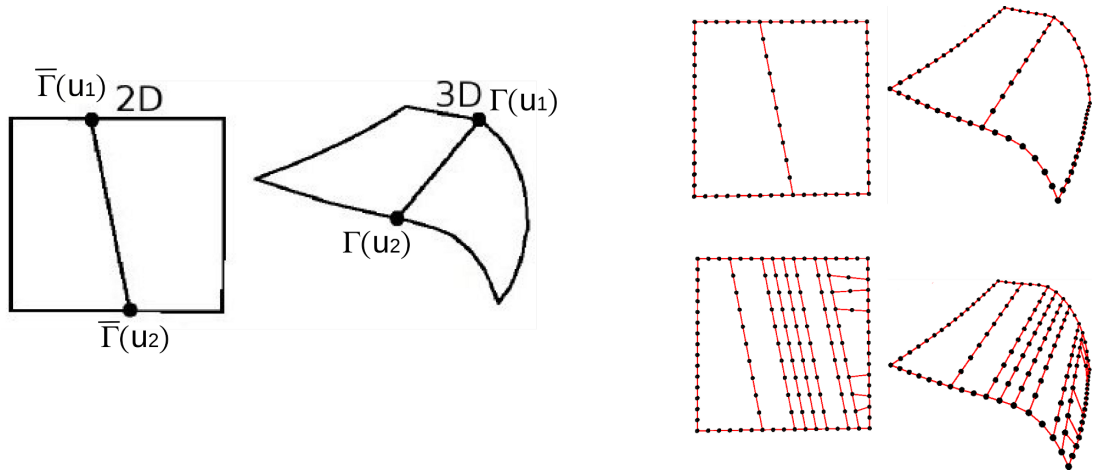


FIGURE 4.8 – *Tracé de segments tendus entre deux positions du bord. la figure de gauche illustre le cas de la correspondance entre le segment sur le patron et la partie 3D. La figure de droite illustre le cas discret suivant un nombre fini de sommets, ainsi que la poursuite de l'algorithme sur ce cas.*

Cas de courbes 3D

Courbe de pli cubique Dans le cas où aucun segment ne peut être tracé entre deux positions du bord, nous cherchons à définir une courbe 3D c entre la paire de sommets dont la distance est la moins compressée comme illustré en fig. 4.9.

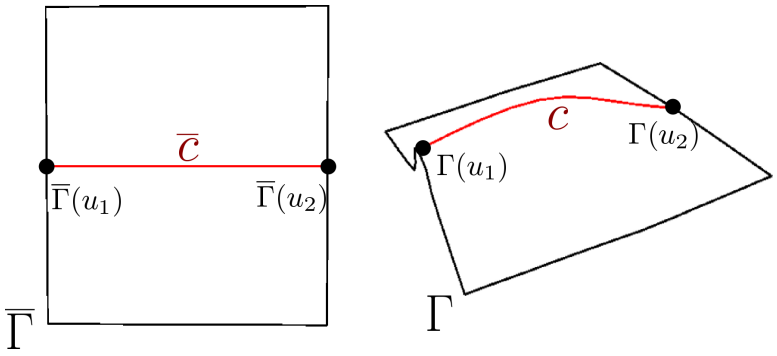


FIGURE 4.9 – *Courbe 3D reliant deux sommets dont la distance est la moins compressée dans le cas de papier plissé.*

Nous choisissons pour cela la courbe c comme étant un polynôme cubique. En effet, les cubiques possèdent les avantages suivants :

- Elles permettent d'approximer des portions de cylindres et de cônes qui forment les patchs de surface C^2 développables.
- Elles minimisent l'intégrale de courbure et empêchent ainsi l'introduction non contrôlée d'oscillations.

– Une interpolation efficace est possible grâce au faible nombre de degrés de liberté de sa paramétrisation.

Les critères de sélection d'une courbe c admissible consistent à trouver une courbe 3D permettant d'améliorer la préservation de l'isométrie lors des étapes de subdivision récursives. Ainsi, nous cherchons à définir la courbe cubique optimale telle que la surface obtenue après subdivision minimise l'erreur de longueur par rapport au patron.

Mesure d'erreur en isométrie Pour cela, nous introduisons une mesure d'erreur en isométrie robuste E entre une surface originale $\overline{S}$ et une surface finale S telle que

$$E(\overline{S}, S) = \sum_{\text{triangles i}} \mathcal{A}_i \ \left((\lambda_1^i - 1)^2 + (\lambda_2^i - 1)^2\right) \ , \tag{4.6}$$

avec $(\lambda_1^i, \lambda_2^i)$ étant les valeurs propres du tenseur d'élongation de chaque triangle i de S par rapport $\overline{S}$ défini au chap. 3.1.1, à l'eq. (3.2). C'est-à-dire que les $\lambda_i - 1$ représente les élongations et contractions propres de chaque triangle dans leur repère local. $\mathcal{A}_i$ représente l'aire du triangle i de $\overline{S}$.

On notera que $E(\overline{S}, S) = 0$ si et seulement si $\overline{S}$ et S sont exactement isométriques, ce qui signifie que S est parfaitement développable.

On notera que cette mesure de préservation de longueur est robuste dans le sens où pour une surface donnée, l'erreur obtenue est indépendante de la connectivité de sa triangulation. Ainsi, $E(\overline{S}, S) = E(\overline{S'}, S')$, avec S' une surface obtenue par subdivision[2] de S. Notons que cette indépendance à la triangulation de la surface ne serait pas réalisée dans le cas d'une estimation d'erreur plus simple telle que la mesure d'élongation sur chaque arête des triangles comme celle proposée à l'eq. (1.31) en partie 1.4.3.

Choix de la cubique optimale Étant donnée la mesure d'erreur en longueur choisie, nous cherchons dans la suite à trouver la courbe de pli permettant de maximiser la préservation d'isométrie.

Soit $(\overline{S_0}, S_0)$ les maillages de l'étape actuelle s'appuyant uniquement sur les bords respectifs $(\overline{\Gamma}, \Gamma)$. Ces maillages peuvent être définis par une triangulation de Delaunay dans l'espace du patron, ils sont donc sans sommets intérieurs.
Soit $(\overline{S}_1(\overline{c}), S_1(c))$ les maillages obtenus après ajout de la courbe de séparation interne c et de son segment associé dans l'espace du patron $\overline{c} = [\overline{\Gamma}(u_1), \overline{\Gamma}(u_2)]$.
Un exemple de tels maillages est illustré en fig. 4.10.

On notera qu'il est nécessaire d'échantillonner de manière cohérente la courbe cubique et le patron 2D. Similairement à la paramétrisation développée en section 4.1, nous échantillonnons la courbe cubique nouvellement introduite suivant son abscisse curviligne relative afin de placer les sommets de la surface s'y appuyant. Notons que l'abscisse curviligne d'une courbe cubique n'est pas exprimable analytiquement et que celle-ci doit être approximée numériquement. Gravesen [Gra92] propose une estimation récursive de la longueur d'un arc de Bézier. Cependant, après compararaison avec une méthode non récursive d'approximation numérique basée sur des étapes de recherches de racines par la méthode de Newton associé, il nous est apparu que la seconde solution était obtenue plus rapidement en pratique. Nous détaillons celle-ci en annexe D.

2. la subdivision doit se réaliser sans modification de géométrie.

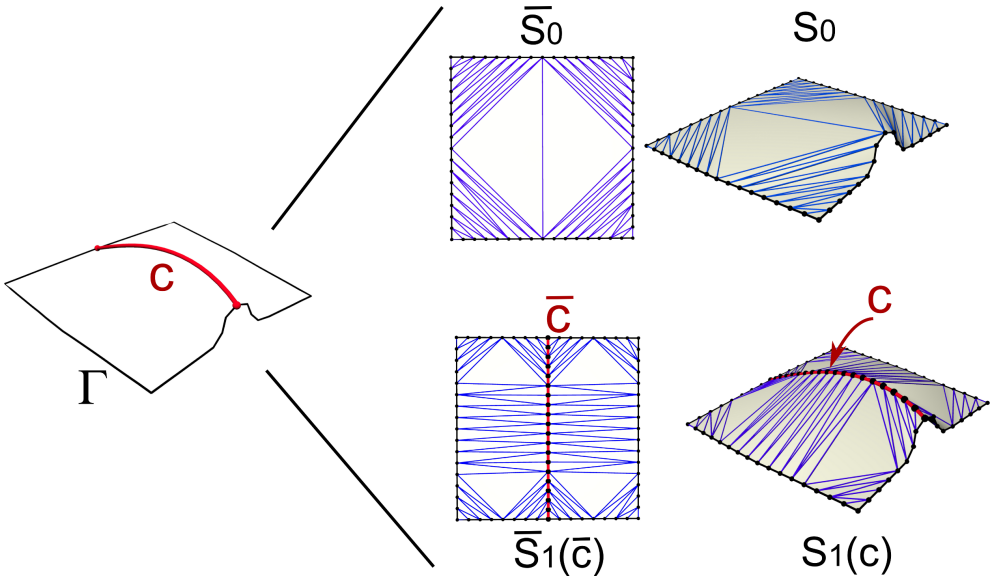


FIGURE 4.10 – *Calcul d'erreur en isométrie sur les maillages s'appuyant sur les courbes avant et après ajout de la courbe de séparation c.*

Une fois la courbe ainsi discrétisée, nous nous intéressons à déterminer la courbe la plus appropriée.

Trouver la surface la plus isométrique au patron peut s'exprimer sous la forme d'un problème de minimisation locale en recherchant la courbe cubique c tel que l'énergie $E(\overline{S}_1(\overline{c}), S_1(c))$ soit minimale. Un exemple d'illustration de différentes courbes possible est proposé en fig. 4.11.

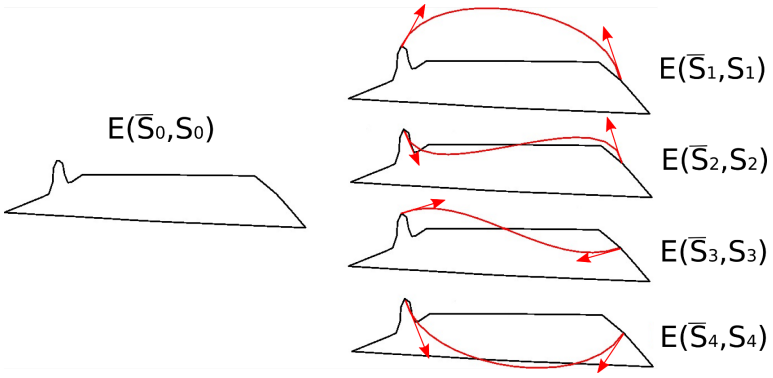


FIGURE 4.11 – *Différentes courbes possibles pour un bord donné. La courbe sélectionnée est celle minimisant l'erreur en isométrie sur les surfaces triangulées associées E par rapport au patron. La courbe finale améliore l'isométrie lors de la subdivision si son erreur associée est inférieure à l'erreur originale issue du couple $(\overline{S}_0, S_0)$.*

Minimisation locale de l'erreur en isométrie Chaque courbe cubique peut être définie par une courbe de Hermite. Les points de départ et d'arrivée étant fixés, les degrés de liberté restants sont donnés par le choix des tangentes aux extrémités. Le choix de la courbe optimale se réalise donc suivant 6 degrés de liberté en 3D.

On notera que l'expression de l'erreur E est non linéaire par rapport aux degrés de liberté. Nous réalisons l'optimisation à l'aide d'une méthode de descente de gradient de type quasi-Newton le long d'une direction donnée par la méthode de BFGS [QSS00]. L'algorithme est initialisé en choisissant c comme étant le segment $[\Gamma(u_1), \Gamma(u_2)]$.
On notera que bien que cette étape soit la plus couteuse de l'algorithme de la génération de la surface, il s'agit d'un sous problème de faible dimension comparé à une optimisation directe réalisée sur l'ensemble des sommets d'un maillage complet. De plus, cette étape n'est mise en oeuvre que si la région possède un pli.

Une fois la courbe optimale obtenue en sortie de l'optimiseur, le gain en isométrie apportée par la subdivision est quantifié par le terme

$$g = \frac{E(\overline{S}_0, S_0)}{E(\overline{S}_1(\overline{c}), S_1(c))} .$$

- Si $g \geq 1$, alors l'ajout est bénéfique et l'isométrie est améliorée (au pire inchangée). L'algorithme récursif peut alors se poursuivre.
- Si $g < 1$, l'ajout de la cubique optimale apporte une diminution de la qualité du respect des longueurs par rapport à une triangulation sans point intérieur.

Le denier cas peut se rencontrer si la surface sous-jacente que l'on tente de modéliser possède plusieurs points d'inflexions. L'approximation par une cubique sera alors de mauvaise qualité. Cette région est alors laissée inchangée à ce stade de l'algorithme et taguée comme incomplète.

Nous présentons dans la partie suivante une approche de relaxation globale permettant de poursuivre l'algorithme lorsque ce cas est rencontré.

4.2.3 Relaxation globale lorsque l'approche locale échoue

L'algorithme présenté précédemment possède deux limitations principales l'empêchant de converger vers la préservation d'isométrie exacte : Premièrement, l'espace des solutions admissibles est uniquement formé par des polynômes cubiques approximant mal le cas de plis complexes. Deuxièmement, l'algorithme procède de manière gloutonne en convergent vers les premiers minimums locaux trouvés. La solution globale obtenue est un minimum local qui ne permet pas de converger vers $E = 0$.

Pour améliorer les propriétés de convergences, nous proposons d'introduire une relaxation globale permettant à la fois d'agrandir l'espace des solutions possibles au-delà des courbes cubiques tout en améliorant la préservation d'isométrie suivant une mesure globale de moindres carrés.

Principe de la relaxation globale

On considère le maillage obtenu après l'ensemble des étapes précédentes. Les sommets de ce maillage sont alors déplacés au sens d'une préservation de rigidité introduite par Sumner et Popovic [SP04] afin de préserver au mieux l'orientation des triangles existants tout en assurant

une déformation la plus isométrique possible entre le patron et le maillage 3D.
Pour cela, on considère la partie rotationnelle [3] des matrices de transformation entre les triangles du patron et les triangles 3D. L'application de ces déformations par triangles définie une soupe de polygone non continue. Ces positions sont alors *recollé* en approximant ces orientation de triangles souhaités au sens des *moindres carrés*. Cette solution étant obtenue comme solution d'un système linéaire creux (voir [SP04]).
Nous obtenons ainsi une déformation appliquée par sommet (voir fig. 4.12) limitant les extensions et compressions par rapport au patron. D'un point de vue complexité, l'approche consiste en une inversion de matrice linéaire creuse et n'influe que très peu sur le temps de calcul.

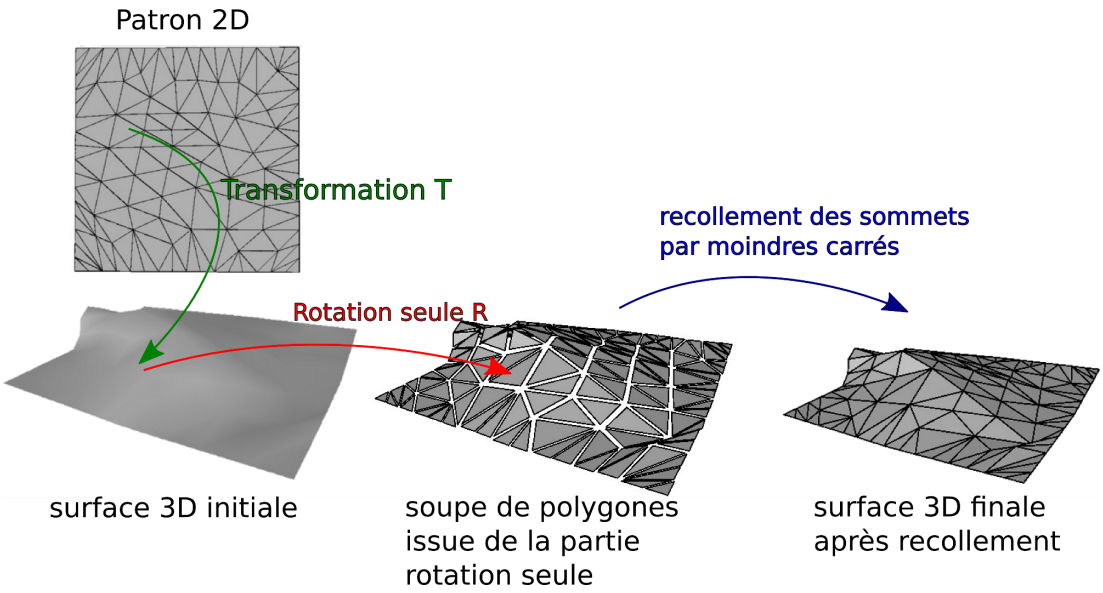


FIGURE 4.12 – *Algorithme de déformation par préservation de rigidité. Les matrices de transformation du patron vers la surface 3D sont exprimées. En ne gardant que la partie rotationnelle des matrices de déformations, nous obtenons une soupe de polygones. Enfin, les sommets sont recollés afin de former un maillage connexe à l'aide d'une minimisation par moindres carrés.*

En pratique, nous observons généralement une amélioration de l'erreur en isométrie de l'eq. (4.6) lors de cette étape. Cette approche permet également de relaxer les contraintes sur les bords originaux. Il est donc possible, si souhaité, de déformer ceux-ci. Notons que cela permet une plus grande robustesse de l'algorithme qui permet de considérer comme entrée un bord Γ non parfaitement isométrique à $\overline{\Gamma}$. Dans ce cas, le bord Γ sera modifié de façon cohérente avec le maillage intérieur pour devenir de plus en plus isométrique (un exemple sera détaillé en partie 4.3.5).

Poursuite de l'algorithme

Une fois le maillage déformé calculé, nous reconsidérons les sommets délimitant les régions taguées comme incomplètes lors de l'algorithme initial. Ces régions délimitées par leurs

3. décomposition matricielle dite *polaire*.

bords discrets servent alors de nouveau point de départ pour l'algorithme récursif comme illustré en fig. 4.13. L'algorithme de génération de plis se termine sur un morceau de surface donné

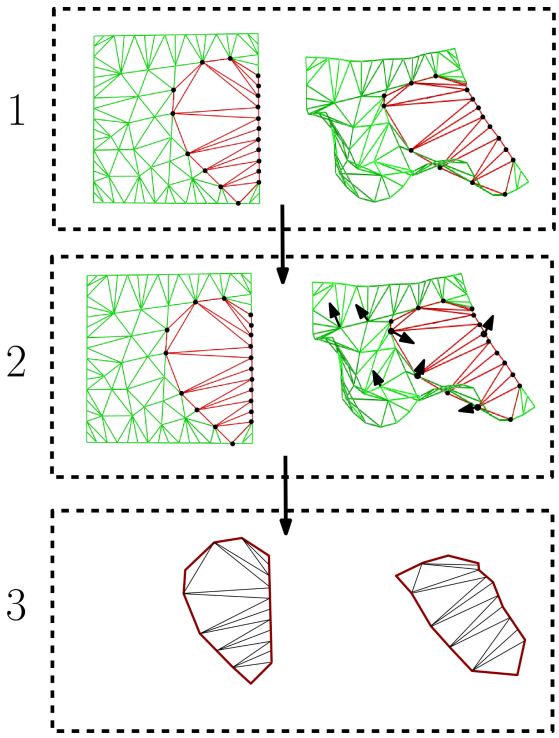


FIGURE 4.13 – *Méthode de relaxation globale. Gauche : vue du patron 2D, Droite : vue 3D associée. 1 : Région rouge indicée comme incomplète, alors que les régions vertes ont convergé vers une préservation d'isométrie correcte. 2 : Déplacement de certains sommets du maillage suivant la méthode de déformation aussi rigide que possible de [SP04]. 3 : La nouvelle région marquée en rouge sert de point de départ pour un nouvel algorithme de subdivision récursive.*

lorsque

- La région est plane
- Les plis ajoutés ont une longueur inférieure à l'échelle de discrétisation utilisée.
- Malgré la relaxation globale précédente, aucune cubique ne parvient à faire décroître l'erreur par rapport à l'état actuel.

Notons que dans le troisième cas, il reste possible d'accepter la *moins mauvaise* cubique afin de poursuivre l'algorithme jusqu'à rencontrer les deux premiers critères d'arrêts. Dans ce cas, bien que n'améliorant pas l'isométrie à l'étape suivante, on observe en pratique un bon comportement de l'algorithme par la suite avec, au final, une diminution globale de l'erreur sur plusieurs étapes.

4.3 SURFACE DE PAPIERS FROISSÉS ET QUANTIFICATION DE LA PRÉSERVATION D'ISOMÉTRIE

4.3.1 Construction de la surface

Une fois que l'algorithme de subdivision est terminé, chacune des régions définies par des bords locaux est approximée la triangulation de celui-ci. Une surface triangulée interpolant les contours initiaux ainsi que l'ensemble des courbes introduites est alors générée. Pour cela, une triangulation de Delaunay contrainte est réalisée dans l'espace du patron. Les segments ajoutés sont utilisés comme contrainte dans l'algorithme de Delaunay afin d'assurer que les arêtes de la surface s'appuient bien sur les plis détectés par l'algorithme. Enfin, les triangles 2D sont alors directement appliqués en 3D étant donné que la bijection de sommets entre ceux du patron et de l'espace 3D est connue.
Cette construction permet ainsi d'obtenir un maillage possédant un nombre limité de triangles tout en modélisant avec précision les angles vifs de plis qui représentent une caractéristique typique du papier plissé. Un exemple d'une telle construction est illustré en fig. 4.14.

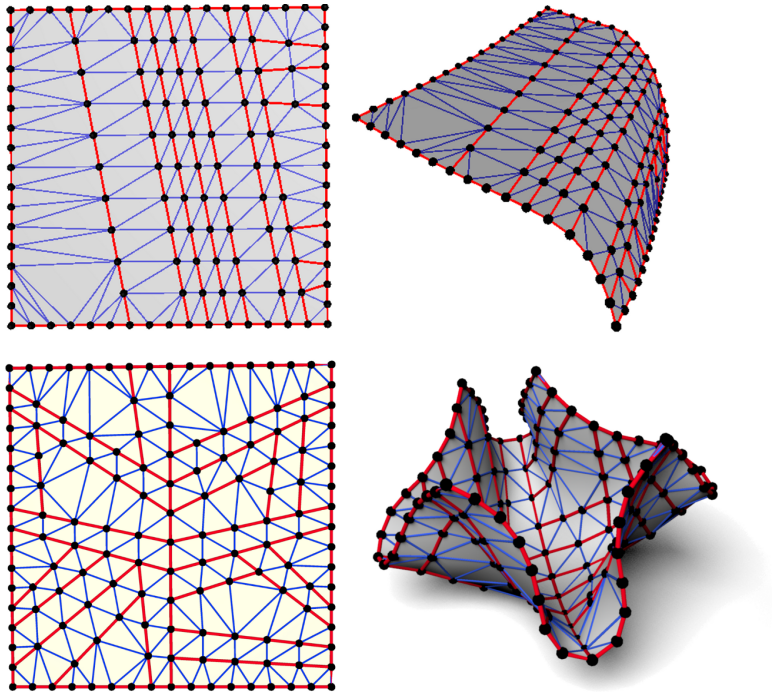


FIGURE 4.14 – *Triangulation pour deux cas de surfaces de papier. Gauche : triangulation dans l'espace du patron. Droite : mapping 3D des triangles.*

4.3.2 Validation sur des surfaces sans plis

Nous procédons à une première validation qualitative des résultats obtenus en appliquant l'algorithme sur des surfaces lisses ne nécessitant pas l'introduction de plis francs.

Les figures 4.15, 4.16 et 4.17 présentent les résultats obtenus sur trois cas différents. Les courbes initiales sont générées par une simulation de tissu très rigide et assure l'existence de surfaces tendues quasi isométriques.

Le premier exemple présente une légère déformation d'un patron rectangulaire. Les lignes de plis principales sont identifiées comme les droites tridimensionnelles, et la géométrie relativement plane est reconstruite rapidement après peu de lignes tirées. Le nombre final de triangles reste donc faible. Le second exemple est un cas de surface dite *strip*. Encore une fois, l'algorithme retrouve les lignes tendues permettant de construire la géométrie ondulante de la bande. Le dernier exemple consiste en une déformation importante de la courbe 3D simulant une feuille tenue par deux extrémités et soumise à la gravité. On notera qu'au moins deux surfaces développables différentes peuvent s'appuyer sur le bord 3D, mais que celle obtenue représente la surface isométrique au patron.

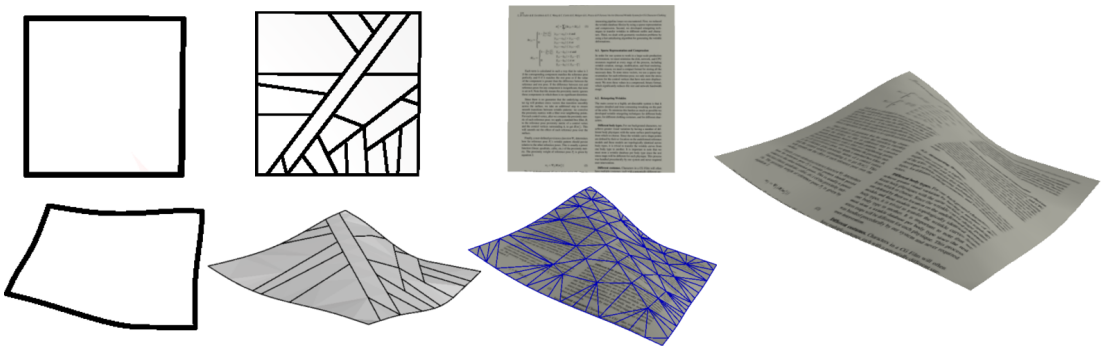


FIGURE 4.15 – *Exemple de reconstruction de surface tendue. De gauche à droite : Courbes initiales (patron 2D et 3D) ; découpage des régions obtenues le long des droites lancées récursivement ; domaine texturé et maillage de la surface 3D ; résultat final.*

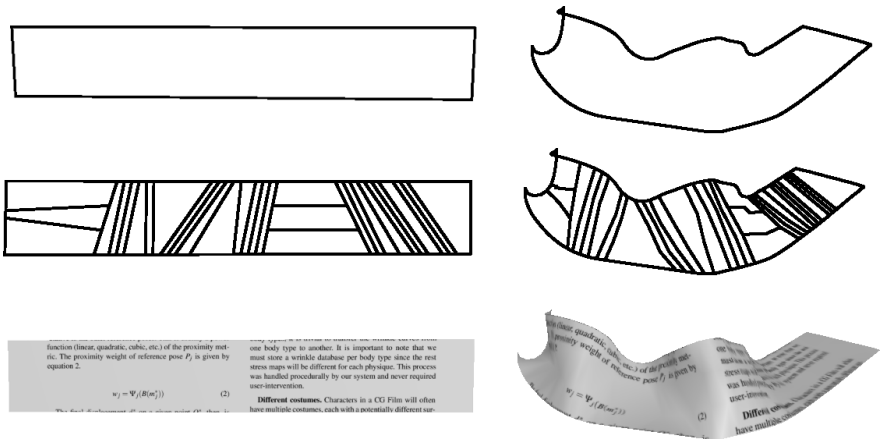


FIGURE 4.16 – *Reconstruction de surface tendue dans le cas d'un bandeau.*

Une observation qualitative montre qu'il n'y a visuellement aucune différence perceptible entre les surfaces construites par notre méthode et les surfaces obtenues par simulation physique de modèles très rigides dont les bords nous fournissent nos données d'entrées.

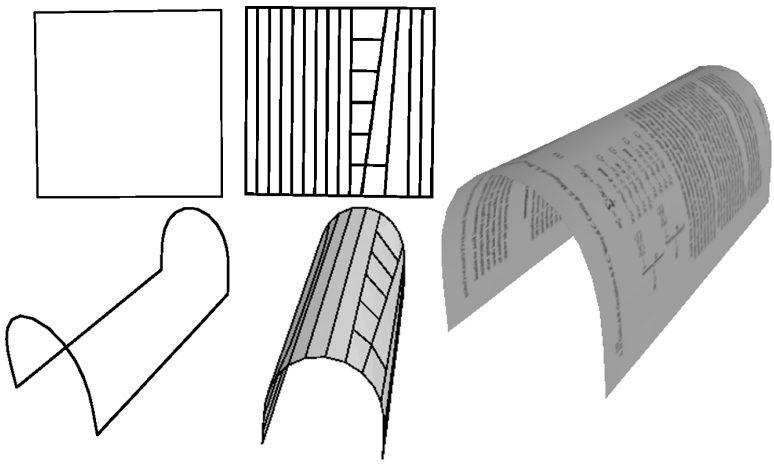


FIGURE 4.17 – *Reconstruction de surface tendue dans le cas d'une déformation 3D importante.*

Des mesures quantitatives rassemblées en fin de document (tableau 4.1) montrent que le respect de l'isométrie des angles est légèrement amélioré pour nos surfaces résultats que pour les surfaces de références issues du modèle physique par masses-ressorts (confirmant que ce dernier capture difficilement l'inextensibilité).

4.3.3 Résultats sur des surfaces plissées

Nous appliquons notre méthode à différents cas de bords représentant des surfaces globalement plissées.

La figure 4.18 représente le cas d'un contour rectangulaire pincé sur un bord. La forme finale obtenue possède des plis internes et est visuellement plausible par rapport à la déformation d'une feuille de papier. La figure 4.19 compare ce résultat à une simulation physique de tissu interpolant les mêmes bords, de même qu'avec une photo d'une vraie feuille de papier. On notera que la surface obtenue est à la fois cohérente vis-à-vis de la contrainte de bord donnée par la simulation physique et visuellement plus proche du véritable papier. On pourra de plus remarquer que la contrainte d'isométrie forte oblige la feuille de papier à descendre plus bas que ses bords au bout de la pliure. Ce phénomène n'apparaît pas dans le modèle de masse-ressort créant de légères élongations, mais se retrouve bien dans notre résultat. L'étude quantitative résumée au tableau 4.1 montre que l'isométrie du maillage obtenu par notre méthode est globalement mieux préservée que par le modèle physique.

La figure 4.20 présente le cas d'une déformation plus importante du bord 3D. La surface sous-jacente est fortement plissée et possède une géométrie sortant largement de l'enveloppe convexe du bord 3D initial. Notre algorithme parvient à donner un résultat plausible visuellement vis-à-vis de la déformation d'une feuille de papier comme montré à droite de la figure. De plus, il est également très proche de la surface obtenue par la simulation physique prenant en compte une contrainte sphérique interne difficilement prévisible à la simple vue du bord original.

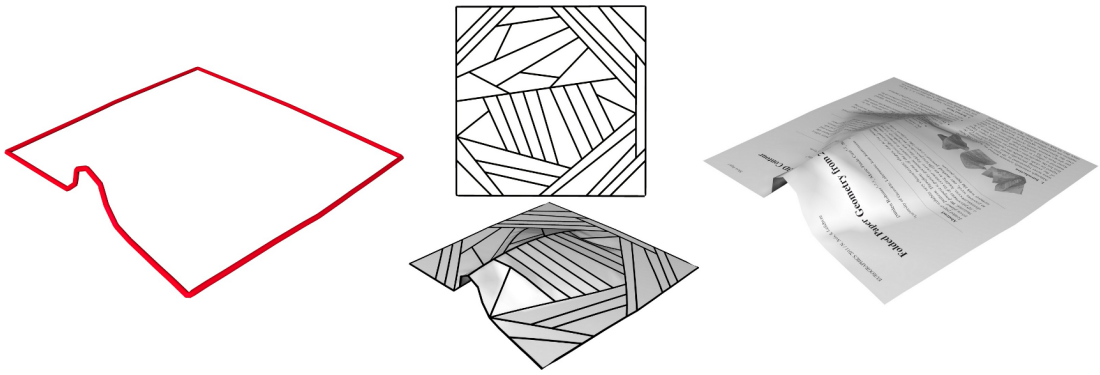


FIGURE 4.18 – *Résultat obtenu pour un bord rectangulaire pincé. Da gauche à droite : bord 3D original ; résultat des courbes de plis sur le patron et la surface 3D ; surface finale texturée.*

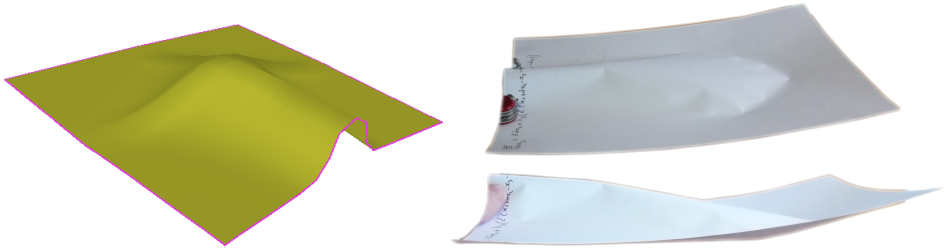


FIGURE 4.19 – *Gauche : Surface obtenue par un modèle masse-ressort de tissu interpolant les mêmes bords que pour notre entrée avec une contrainte de non-pénétration sphérique interne. Droite : Photos d'une vraie feuille de papier pliée sous deux angles différents. Nous remarquons combien le modèle masse-ressorts est une mauvaise approximation du papier.*

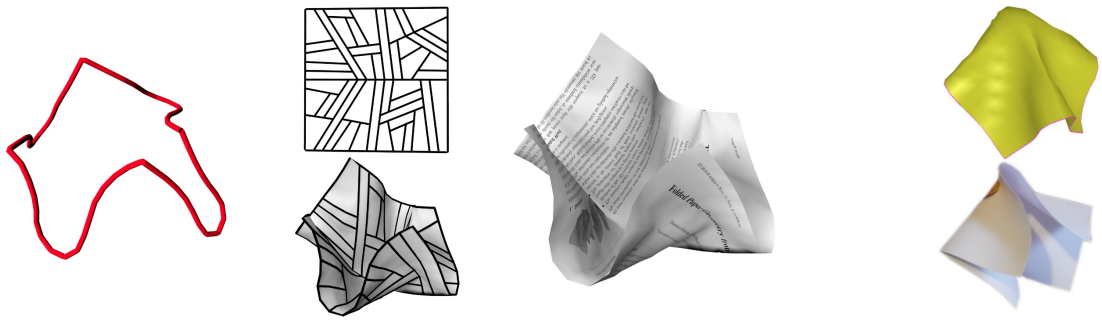


FIGURE 4.20 – *Résultat obtenu dans le cas d'un bord fortement plissé. La comparaison avec une simulation masse-ressorts (jaune) ainsi qu'avec une vraie photo est également montrée à droite.*

Un exemple en fig. 4.21 montre le cas d'une surface dont les plis ne sont pas modélisables par des courbes planes. Nous rappelons que nous ne réalisons aucune supposition sur une direction privilégiée de déformation, la contrainte forte d'isométrie permettant à elle seule de retrouver des directions plausibles.

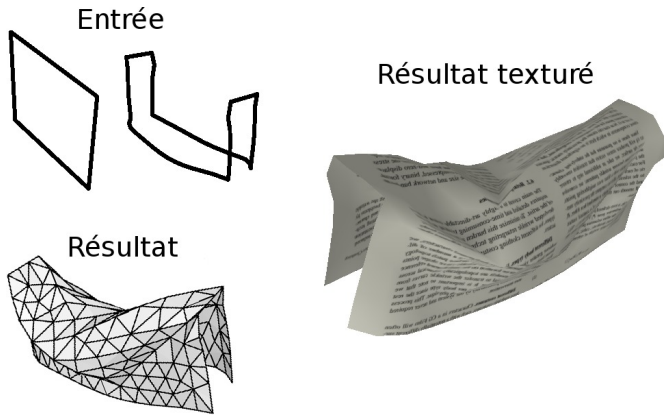


FIGURE 4.21 – *Surface quasi-isométrique générée à partir de plis non plans.*

La fig. 4.22 illustre d'autres exemples de papiers froissés montrant que l'approche peut s'adapter à des bords 3D variés.

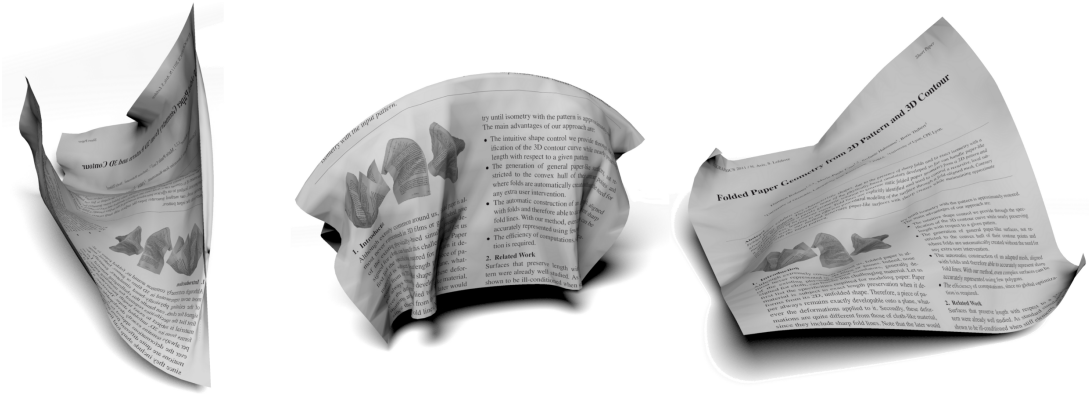


FIGURE 4.22 – *Différents résultats de formes de papiers pour divers bords 3D.*

Enfin, nous pouvons étendre notre algorithme à la modélisation d'un matériau autre que du papier. Ainsi, la fig. 4.23 présente une application au cas de surface en métal formée par une canette compressée. Dans ce cas, le bord 3D original est issu d'une simulation par masse-ressort réalisée sur un cylindre compressé.

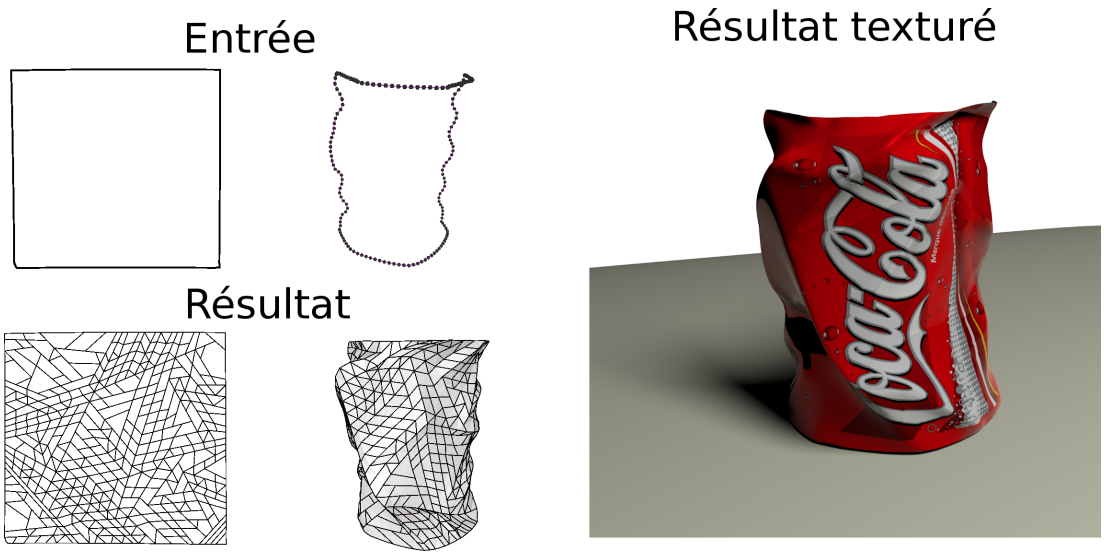


FIGURE 4.23 – *Extension de l'algorithme au cas de surface en métal.*

	E	E_{angle}	$E_{\text{area}} \times 10^4$	tps
fig 4.15	0.21 (0.25)	0.35 (1.90)	1.3 (1.1)	<0.1
fig 4.16	0.09 (0.21)	0.16 (1.40)	0.7 (1.1)	<0.1
fig 4.17	0.56 (0.79)	2.60 (17.6)	5.0 (2.0)	<0.1
fig 4.18	1.28 (2.50)	2.89 (22.8)	8.0 (18)	0.2
fig 4.20	1.28 (2.50)	2.89 (22.8)	8.0 (18)	0.6
fig 4.21	1.50 (8.60)	2.80 (68.7)	8.0 (42)	0.6

TABLE 4.1 – *Mesure d'erreurs et de temps de calcul (en seconde) pour les exemples des figures citées.*

4.3.4 Quantification de l'isométrie

Le tableau 4.1 suivant donne une mesure quantitative du respect de l'isométrie, d'angle et d'aire pour les différents exemples présentés. Le premier chiffre représente le résultat obtenu par notre méthode, et le second entre parenthèses mesure le même critère sur le maillage obtenu en sortie du simulateur physique (simulation de tissu du logiciel Blender).

E est la mesure d'erreur d'isométrie définie par l'eq. (4.6). E_{angle} représente l'erreur totale en angle entre ceux des triangles plan du patron et le maillage 3D

$$E_{\text{angle}} = \sum_{\text{angle}_j} |\alpha_j - \overline{\alpha}_j| \,,$$

avec α_j (resp. $\overline{\alpha}_j$) le $j$ième angle du maillage 3D (resp. patron 2D). Enfin E_{area} représente l'erreur finale en aire sommée sur l'ensemble des triangles.

$$E_{\text{area}} = \sum_{\text{triangle}_k} |\mathcal{A}_k - \overline{\mathcal{A}}_k| \,,$$

où $\mathcal{A}_k$ (resp. $\overline{\mathcal{A}}_k$) correspond à l'aire du $k$ième triangle de la surface 3D (resp. 2D). Le calcul de ces deux erreurs est illustré en fig. 4.24

On notera que dans l'ensemble de nos exemples, le patron est normalisé de telle sorte que son côté le plus long ait une longueur unitaire.

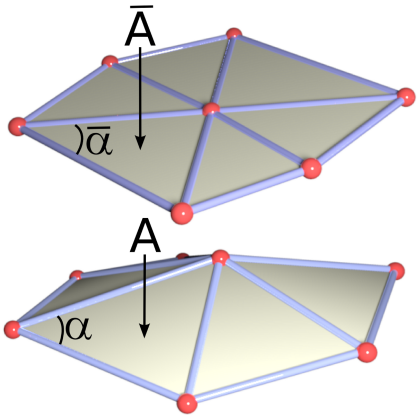


FIGURE 4.24 – *Mesure de l'erreur en angle et en aire entre les triangles du patron (haut) et 3D (bas).*

La dernière colonne mesure le temps de calcul qui bien que loin de permettre une modélisation de plis isométriques en temps-réel, reste inférieur à celui des méthodes de minimisation globales.

4.3.5 Extension au cas de surfaces non développables

L'algorithme de construction de surface que nous avons proposé fonctionne par vérification d'une suite d'inégalité. Il est spécifiquement défini afin de pouvoir traiter de manière robuste

les cas où les bords 3D ne sont pas compatibles avec la génération d'une surface préservant l'isométrie vis-à-vis du patron.

La fig. 4.25 montre l'application de notre algorithme dans le cas où le bord 3D est artificiellement compressé et artificiellement agrandis. Dans le premier cas, on peut noter que la compression viens engendrer des plis tendant de préserver au mieux l'isométrie. Bien que la préservation exacte des longueurs soit ici impossible, les plis engendrés restent visuellement plausibles. Dans le second cas de l'extension, l'algorithme n'ajoute aucun pli interne et se contente de générer la *moins mauvaise* surface interpolant les bords qui consiste en la triangulation de Delaunay formée par les sommets du contour.

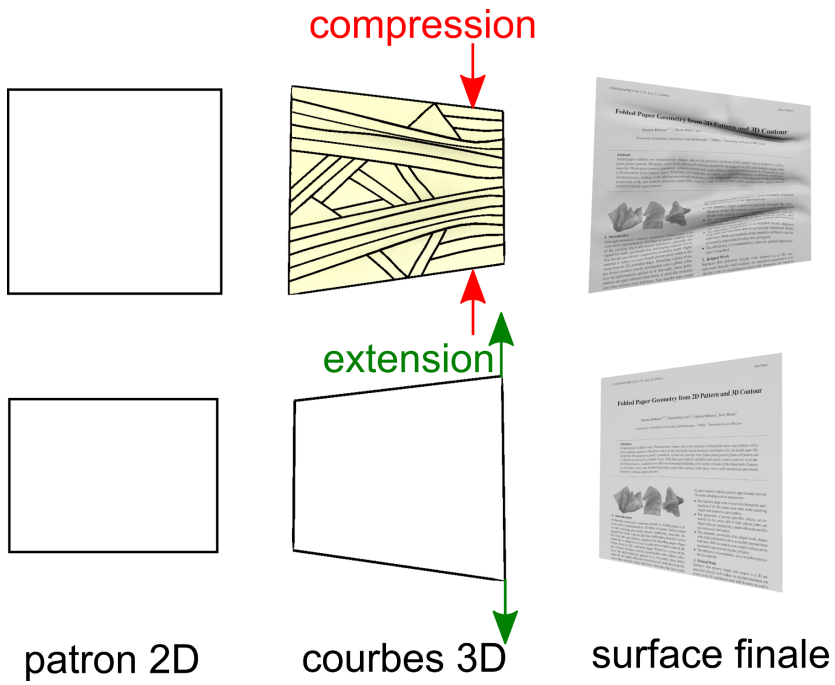


FIGURE 4.25 – *Application de l'algorithme de génération de surfaces pour des bords 3D arbitrairement compressés ou étendus.*

On notera également que l'utilisation de l'algorithme de relaxation globale permet de modifier le bord 3D de la surface de manière à améliorer un bord non initialement compatible avec une surface développable. La courbe 3D donnée en tant que paramètre peut ainsi être améliorée afin de mieux préserver l'isométrie au cours de la reconstruction. Un exemple d'une telle modification est illustré en fig. 4.26 où l'on peut visualiser le bord final de la surface après déformation par rapport au bord original non exactement isométrique au patron. On notera que dans tout les cas, il est toujours possible de contraindre le bord à rester identique à celui donné en entrée en forçant une contrainte stricte de position sur les sommets du bords lors de l'étape de relaxation.

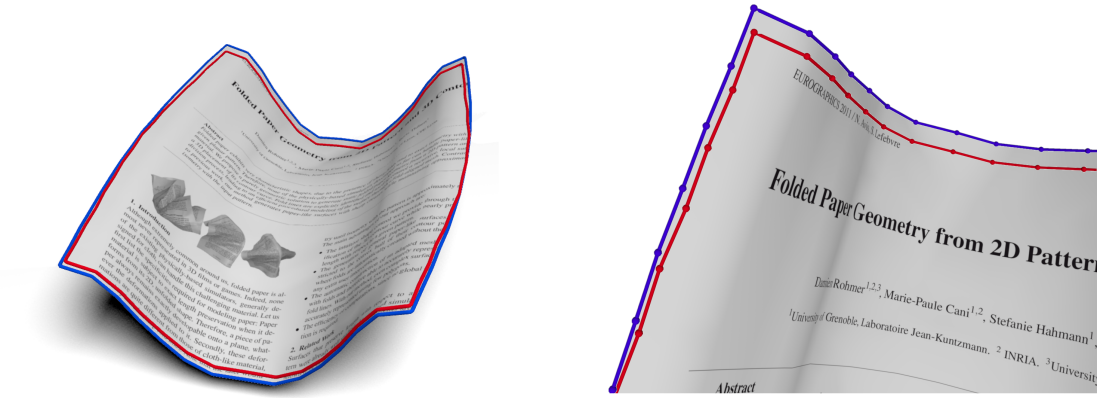


FIGURE 4.26 – *Déformation du bord 3D de la surface lors de l'étape de relaxation globale. Le bord initial donné en paramètre est indiqué en rouge, alors que le bord final après convergence est donné par la courbe bleue.*

4.4 LIMITATIONS ET TRAVAUX FUTURS

L'algorithme de génération de surface quasi développable permet de modéliser des formes plausibles de papier froissé. Cependant, elle possède un certain nombre de limitations qui ouvrent la voie à de futurs travaux de recherche.

Premièrement, la nécessité de générer un bord 3D compatible avec un patron 2D donné est complexe et ne permet pas la modélisation intuitive d'une forme générique. Nous avons présenté un algorithme de déformation de courbes tentant de préserver la développabilité en partie 4.1. Cependant, celui-ci ne permet de modéliser qu'un ensemble limité de déformations et n'apporte pas la liberté qu'un artiste pourrait souhaiter. Il s'en suit que la génération de bords 3D complexes a été réalisée lors de précalculs par simulateur de tissus, la forme de la courbe du bord obtenue n'étant alors pas directement contrôlable. Le contrôle direct d'un échantillon de quelques sommets intérieurs permettrait un contrôle plus aisé de la forme. Reconstruire la forme finale à partir de la connaissance des points de contact des doigts d'une main avec la feuille de papier permettrait une modélisation plus intuitive de telles formes comme illustrée en fig. 4.27.

Nous pourrons également noter que la génération de surface proposée ne converge en général pas vers une surface parfaitement isométrique. La méthode de relaxation linéaire globale permet de limiter l'impact des minima locaux, mais ne permet pas d'assurer de convergence. Néanmoins, il est raisonnable d'estimer qu'une approximation de la préservation d'isométrie est suffisante pour obtenir un résultat visuel convaincant.

Une autre limitation actuelle concerne l'absence de gestion de collisions, à la fois avec des éléments extérieurs ainsi que les autocollisions elles-mêmes. Au-delà de l'apparence irréaliste des autocollisions pouvant intervenir dans certaines configurations, il n'est également pas possible dans la version actuelle de modéliser un papier fortement froissé. En effet, la forme d'un tel papier est alors largement guidée par la non-pénétration de la surface sur elle-même.

Considérer une déformation continue lors d'une animation temporelle serait également une piste intéressante de poursuite de cette étude. La reconstruction indépendante à chaque pas de temps a déjà fait l'objet de tests comme montré en fig. 4.28 et permet la génération d'une suite de surfaces interpolant les bords animés. L'ajout d'une cohérence temporelle et la prise

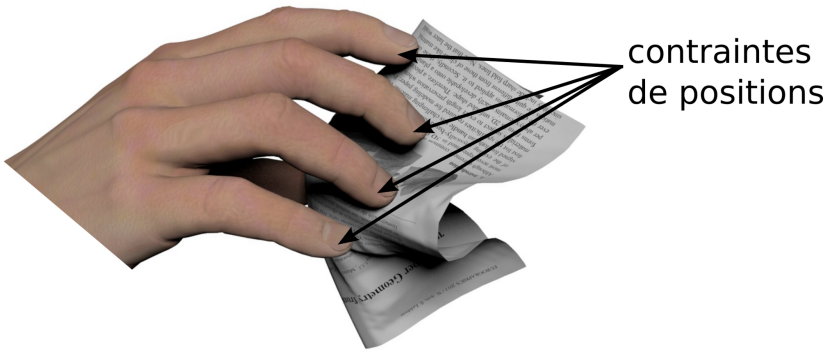


FIGURE 4.27 – *Il serait souhaitable d'autoriser la donnée de contraintes ponctuelles de position, comme dans cet exemple fictif (ici la main à été placée une fois le modèle de papier calculé).*

en compte du comportement irréversible de la formation de plis francs permettraient d'obtenir des animations convaincantes de papier froissé.

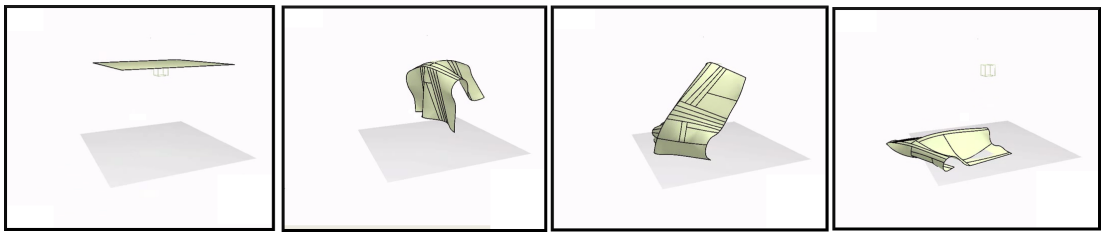


FIGURE 4.28 – *Animation d'une surface quasi-developpable obtenue par une série de reconstructions indépendantes au cours du temps.*

Afin d'améliorer le modèle de surface, l'ajout de propriétés physiques telles que la prise en compte du comportement anisotrope induit par la structure microscopique du papier permettrait d'entrevoir la génération d'un modèle visuel rapide et réaliste de papier animé.

BILAN DE LA MODÉLISATION DE PAPIER FROISSÉ

Nous avons présenté une méthode de génération rapide de surface quasi-isométrique à un patron donné interpolant le bord d'une courbe 3D.
L'approche permet d'obtenir des résultats visuels ressemblant à du papier froissé et ouvre la voie à la modélisation de telles surfaces encore peu repandue dans le domaine graphique.

Nous avons tout d'abord développé une méthode de déformation d'un bord 3D tout en préservant certaines propriétés de développabilité. La génération de la surface elle-même a ensuite été présentée suivant une approche géométrique locale permettant de subdiviser la contrainte de préservation de longueur en sous problèmes de dimensions plus faible. Enfin, nous avons introduit une relaxation globale permettant d'améliorer le respect de l'isométrie tout en autorisant l'ajustement robuste des bords dans le cas de données d'entrée non compatibles avec l'interpolation par une surface isométrique.

Les résultats obtenus permettent de donner une impression visuelle convaincante de formes de papier tout en offrant une préservation quantitative d'isométrie suffisante, et notamment meilleure qu'une simulation de type masse-ressort rigide interpolant les mêmes bords. Enfin, l'approche géométrique locale permet d'obtenir un résultat de manière quasi interactive (<1s), ce qui constitue une amélioration claire par rapport aux méthodes d'optimisation globales.

Ce travail ouvre la voie à diverses possibilitées de poursuites dans l'espoir de pouvoir converger vers un outil robuste de modélisation de surfaces de papier animables, et plausibles.

CHAPITRE

5

BILAN ET PERSPECTIVES

Nous avons présenté dans cette thèse différentes approches permettant d'animer et de modéliser des objets géométriques dont certaines propriétés intrinsèques sont conservées une fois plongées dans l'espace 3D. Les méthodes générant ces objets, qui réagissent de manière *active* pour préserver leurs contraintes structurelles, permettent d'améliorer la plausibilité tout en déchargeant l'artiste de ces aspects pour la création de mondes virtuels.

Le but des approches décrites dans ce document était de définir ces modèles actifs tout en considérant quatre critères principaux d'utilisabilité. Voici la réponse que nous avons proposée à chacun de ces critères :

- **Plausibilité** : La préservation des caractéristiques géométriques intrinsèques aux différents modèles a permis d'améliorer la plausibilité de son apparence par rapport à l'utilisation d'une méthode standard de déformation. Nous avons pour cela présenté :
  i. Des modèles de personnages se déformant à volume constant ou contrôlé localement, générant ainsi des gonflements locaux lors de leurs articulations.
  ii. L'ajout de plis géométriques de tissu permettant d'améliorer l'impression de dynamique tout en limitant les compressions irréalistes.
  iii. Une génération de surfaces de papier froissé basée sur la préservation d'isométrie vis-à-vis d'un patron et garantissant une apparence non lisse typique.

- **Rapidité** : L'extraction de relations géométriques exprimées analytiquement visant à satisfaire les contraintes nous a permis de proposer des solutions fonctionnant à vitesse interactive. Nous nous sommes ainsi basés sur ces différentes approches :
  i. Un post process rapide de déformation par sommet appliqué sur une déformation sous-jacente par skinning interactive.
  ii. Un post process d'ajout de détails géométriques, ajouté à une simulation à basse résolution, l'ensemble étant plus rapide qu'un calcul complet à haute résolution.
  iii. Une réduction de dimensionalité d'un problème de préservation de longueur non linéaire aboutissant à un procédé constructif rapide.

– **Contrôle** : Il s'agissait d'offrir des paramètres de contrôle de haut niveau, simples et visuels, et ne requièrant pas le jeu d'essais/erreurs classiques des approches physiques standard. Les contrôles que nous avons proposés se sont réparti de cette manière :
  i. Une manipulation de gonflements locaux guidés par le tracé de courbes de profils permettant la mise en place d'effets de muscles ou de bourrelets.
  ii. La définition des types de plis de vêtement par la manipulation de l'épaisseur du tissu indépendamment de la dynamique sous jacente de celui-ci.
  iii. La manipulation directe d'un bord d'une surface de papier.

– **Utilisabilité** : L'intégration des méthodes proposées dans des pipe-lines standard d'animation permet leur utilisation sur des données préexistantes. Nous avons pour cela utilisé les méthodes suivantes :
  i. L'articulation globale d'une forme par skinning.
  ii. La déformation de base d'un vêtement par un modèle masse-ressorts à basse résolution.
  iii. La modélisation d'une surface à partir d'une courbe de bord 3D.

Les résultats principaux obtenus ont permis d'animer des personnages de manière plausible en simulant des étirements, des effets de muscles ou de bourrelets dans un contexte interactif, et ce, à partir de la simple donnée initiale d'un maillage et d'un squelette d'animation. Nous avons également pu générer des animations de tissus visuellement réalistes grâce à l'ajout de détails géométriques dont la forme et l'orientation s'adaptent automatiquement à une animation quelconque à basse résolution sous-jacente. Enfin, des formes visuellement convaincantes de papier froissé ont pu être générées interactivement tout en garantissant une préservation d'isométrie approximée vis-à-vis d'un patron donné.

Les objets géométriques *actifs* que nous avons proposés sont conçu pour être intégrés dans des scènes virtuelles interactives. À ce titre, de nombreuses possibilités de poursuites futures sont possible afin à la fois d'enrichir la diversité des propriétés de ces objets, ainsi que l'amélioration de leur réalisme visuel, contrôle utilisateur et efficacité de calcul. Nous noterons ainsi que l'ensemble des méthodes proposées dans ce document aurait avantage à tirer parti d'implémentations optimisées sur architecture parallèle telles que les cartes graphiques. La plupart des méthodes proposées ici pourraient ainsi être calculées en temps réel et directement être intégrées dans des jeux vidéos.

Nous avons pu également constater que l'intégration combinée de la préservation de certaines caractéristiques géométriques permettait de simuler des effets plus complexes. Par exemple, la combinaison entre la déformation articulée à volume constant présentée au chapitre 2 combinée avec la génération de plis pour de la peau du chapitre 3 permettrait d'améliorer le réalisme de tels personnages.

Plus généralement, nous souhaiterions tendre à terme vers la définition d'un système de création et d'animation d'objets virtuels permettant aux utilisateurs de se consacrer à leur choix artistique. La mise en place d'un cadre de travail basé sur des surfaces actives permettrait ainsi de décharger la gestion des détails à compléter ou le respect de certaines contraintes importantes au système. Un contrôle intuitif sur la forme et l'animation finale restant accessible en permanence à l'aide d'un paramétrage adapté et mesurable dans le monde réel.

Premièrement, l'incorporation de nouvelles propriétés géométriques à préserver et liées à la structure interne des matériaux telle que la préservation de courbures moyenne pour la

modélisation interactive de surfaces minimales nécessaire pour la modélisation de bulles de savon [ZYP06] ; la prise en compte d'anisotropie liée aux structures fibrées présente dans les végétaux [JK10] ou les organes [TAI+08] ; ou encore la modification de la métrique de la surface obtenue lors du déchirement de matière plastique [SRS07] permettrait d'enrichir la gamme de modèles pouvant être généré par ce système. La prise en compte de ces propriétés complexes par le système permettrait ainsi à l'utilisateur de déformer ces matières par des outils génériques (déformation basée surface, ou de l'espace) sans avoir à générer manuellement l'ensemble des détails géométriques liés à la constitution physique de l'objet en question.

Deuxièmement, la modélisation d'objets multi-composants nécessite l'incorporation d'a priori sémantiques supplémentaires sur la caractérisation de certains détails et propriétés géométriques tels que l'évitement des auto-intersections (ex. robe formée par couches successives de tissu [ZXTD10]), changement de topologie (ex. contrainte forte sur du papier aboutissant à sa déchirure [KMB+09]), duplication de détails (ex. agrandissement ou rétrécissement d'un animal formé d'écailles [LS10]). Coupler nos modèles de surfaces *active* dans le cadre de telles formes permetterait d'étendre la modélisation efficace d'objets complexes dans les mondes virtuels.

Troisièmement, la mise en place d'interactions avec des simulateurs permetterait de tendre vers l'animation de modèles actifs complexes de réalisme croissants, tout en garantissant un caractère contrôlable intuitif. Par exemple, la prise en compte du changement de rigidité d'un tissu le long des courbes de plis permettrait de modifier dynamiquement la rigidité, et donc le comportement, de la simulation physique [PKST08]. De même que l'intégration des plis francs de papier permet d'entrevoir la génération d'un simulateur de papier tenant compte du caractère irréversible du froissage [GHDS03]. Cette interaction bidirectionnelle entre surface active et modèle physique permettrait ainsi d'accroitre l'efficacité et le domaine d'application des simulateurs standards.

Quatrièmement, l'intégration à des méthodes d'interaction avancées permetterait d'approfondir les possibilités de manipulation de nos modèles. L'esquisse par croquis [TWB+07] de détails tels que le profil de bourrelets ou ceux des plis de tissus directement sur le modèle permettrait un contrôle accru sur la réponse géométrique de nos modèles *actifs*. Également, l'intégration de modalité de sculptures [SCC11] dans le cadre d'un modèle *actif* ouvrirait la voie à la modélisation à l'aide d'outils encore inexplorés tels que la sculpture d'une forme en papier ou en tissu.

Au final, nous espérons que le travail présenté ici puisse représenter un premier pas vers des modèles virtuels convaincants déformables et animables, réagissant de manière autonome, tout en restant contrôlables intuitivement afin de permettre le partage aisé de l'imagination artistique.

ANNEXE

A

CALCUL DU VOLUME DÉLIMITÉ PAR UNE SURFACE

Cette annexe démontre les formules utilisées pour calculer le volume V délimité par une surface S.

A.1 CAS CONTINU

Soit Ω un compact de $\mathbb{R}^3$ délimité par une surface S. Le volume V du domaine Ω est défini par

$$V = \int_{\Omega} \mathrm{d}\Omega \,. \tag{A.1}$$

Le théorème de la divergence (ou théorème de Green-Ostrogradski) sur une fonction vectorielle $\mathbf{f} : \mathbf{p} \mapsto \mathbf{f}(\mathbf{p})$ quelconque C^1 sur le domaine Ω donne

$$\int_{\Omega} \nabla \cdot \mathbf{f}(\mathbf{p}) \, \mathrm{d}\Omega = \int_{\partial\Omega} \left\langle \mathbf{f}(\mathbf{p}), \frac{\mathbf{n}(\mathbf{p})}{\|\mathbf{n}(\mathbf{p})\|} \right\rangle \mathrm{d}(\partial\Omega) \,, \tag{A.2}$$

où $\mathbf{n}$ correspond à la normale sortante de Ω situé au point $\mathbf{p}$ du bord $\partial\Omega$. Considérons le cas particulier où $\mathbf{f}(\mathbf{p}) =< \mathbf{p}, \mathbf{e} > \mathbf{e}$, avec $\mathbf{e}$ vecteur unitaire quelconque. En remarquant que

$$\nabla \cdot < \mathbf{p}, \mathbf{e} > \mathbf{e} = \|\mathbf{e}\|^2 = 1 \,,$$

l'equation (A.2) s'écrit alors [GOCP97]

$$\int_{\Omega} \mathrm{d}\Omega = \int_{S} < \mathbf{p}, \mathbf{e} > \frac{< \mathbf{n}, \mathbf{e} >}{\|\mathbf{n}\|} \mathrm{d}(\partial\Omega) \quad . \tag{A.3}$$

Ce qui, dans le cas d'une surface paramétrée suivant (u, v) permet de retrouver la formule de l'eq. (1.21)

$$V = \iint_{(u,v)\in\mathcal{D}} < \mathbf{p}(u,v), \mathbf{e} >< \mathbf{n}(u,v), \mathbf{e} > \, \mathrm{d}u\,\mathrm{d}v \, ,$$

avec la normale $\mathbf{n}$ telle que

$$\mathbf{n}(u,v) = \frac{\partial \mathbf{p}}{\partial u}(u,v) \times \frac{\partial \mathbf{p}}{\partial v}(u,v) \, .$$

A.2 Cas d'une surface polygonale

Dans le cas où S est représenté par un maillage polygonale à face planaires, on peut alors décomposer cette intégrale en sommant la contribution de chacune des faces. Soit i l'indice d'une face. $\mathcal{F}_i = (k_1^i, k_2^i, k_3^i, \cdots, k_p^i)$ le p-uplet d'indices de la face i de p sommets.

$$V = \sum_{i=1}^{N_f} \left(\iint_{(u,v)\in\mathcal{D}_i} < \mathbf{p}_i(u,v), \mathbf{e} >< \mathbf{n}_i, \mathbf{e} > \, \mathrm{d}u\,\mathrm{d}v \right) \, ,$$

avec $\mathcal{D}_i$ le domaine de définition paramétrique du polygone i, $\mathbf{p}_i$ un point de ce polygone, et $\mathbf{n}_i$ sa normale. Notons que les faces sont planes, et que les normales sont donc constantes sur celui-ci. On a donc

$$V = \sum_{i=1}^{N_f} < \mathbf{n}_i, \mathbf{e} > \left\langle \iint_{(u,v)\in\mathcal{D}_i} \mathbf{p}_i(u,v) \, \mathrm{d}u\,\mathrm{d}v, \mathbf{e} \right\rangle \, .$$

A.3 Cas de polynomes triangulaires

Soit un maillage consitué exclusivement de triangles. On a alors pour tout i, $\mathbf{F}_i = (k_1^i, k_2^i, k_3^i)$. Le triangle i est donc formé des trois positions spatiales $\mathbf{p}_{k_1^i}, \mathbf{p}_{k_2^i}, \mathbf{p}_{k_3^i}$.

Paramétrons le triangle i par rapport au triangle isocèle unitaire. On a alors

$$\begin{cases} \mathcal{D} = \left\{ (u,v) \in \mathbb{R}^2 | u \in [0,1] \text{ et } \; v \in [0, 1-u] \right\} \\ \\ \forall (u,v) \in \mathcal{D} \, , \quad \mathbf{p}_i(u,v) = \mathbf{p}_{k_1^i} + (\mathbf{p}_{k_2^i} - \mathbf{p}_{k_1^i})u + (\mathbf{p}_{k_3^i} - \mathbf{p}_{k_1^i})v \, . \end{cases}$$

D'où

$$\begin{aligned} \iint_{(u,v)\in\mathcal{D}_i} \mathbf{p}_i(u,v) \quad &= \int_{u=0}^{1} \int_{v=0}^{1-u} \mathbf{p}_{k_1^i} + (\mathbf{p}_{k_2^i} - \mathbf{p}_{k_1^i})u + (\mathbf{p}_{k_3^i} - \mathbf{p}_{k_1^i})v \, \mathrm{d}u\,\mathrm{d}v \, . \\ &= \frac{1}{6} \left(\mathbf{p}_{k_1^i} + \mathbf{p}_{k_2^i} + \mathbf{p}_{k_3^i} \right) \, . \end{aligned} \tag{A.4}$$

De plus, la normale du triangle peut s'exprimer par

$$\mathbf{n}_i = (\mathbf{p}_{k_2^i} - \mathbf{p}_{k_1^i}) \times (\mathbf{p}_{k_3^i} - \mathbf{p}_{k_1^i}) \, .$$

Le produit mixte $< \mathbf{n}_i, \mathbf{e} >$peut s'exprimer sous forme d'un déterminant

$$< \mathbf{n}_i, \mathbf{e} >= \det(\mathbf{p}_{k_2^i} - \mathbf{p}_{k_1^i}, \mathbf{p}_{k_3^i} - \mathbf{p}_{k_1^i}, \mathbf{e}) \tag{A.5}$$

En combinant l'eq. (A.4) et (A.5), nous pouvons exprimer l'equation générale du volume d'un maillage triangulé pour tout vecteur $\mathbf{e}$ unitaire

$$V = \frac{1}{6} \left\langle \mathbf{p}_{k_1^i} + \mathbf{p}_{k_2^i} + \mathbf{p}_{k_3^i}, \mathbf{e} \right\rangle \det(\mathbf{p}_{k_2^i} - \mathbf{p}_{k_1^i}, \mathbf{p}_{k_3^i} - \mathbf{p}_{k_1^i}, \mathbf{e}) \, .$$

Il est courant de considérer $\mathbf{e} = \mathbf{u}_z = (0, 0, 1)$, on a alors

$$V = \frac{1}{6} \sum_{i=1}^{N_f} (z_{k_1^i} + z_{k_2^i} + z_{k_3^i}) \begin{vmatrix} x_{k_2^i} - x_{k_1^i} & x_{k_3^i} - x_{k_1^i} \\ y_{k_2^i} - y_{k_1^i} & y_{k_3^i} - y_{k_1^i} \end{vmatrix}$$

A.4 VOLUME ASSOCIÉ À UN CARREAU BILINÉAIRE

Soit $\mathbf{p}_a, \mathbf{p}_b, \mathbf{p}_c, \mathbf{p}_d$ les quatres sommets d'un carraux bilinéaire comme illustré en fig. 2.10. Nous démontrons la relation donnée en eq. (2.1).

Soit

$$f \begin{cases} [0,1]^2 \subset \mathbb{R}^2 & \to & \mathbb{R}^3 \\ (u,v) & \mapsto & f(u,v) \, , \end{cases}$$

la fonction paramétrique associée au carreau bilinéaire. On a donc

$$\forall (u,v) \in [0,1]^2 \, , \quad f(u,v) = (1-u)(1-v)\mathbf{p}_a + u(1-v)\mathbf{p}_b + uv\mathbf{p}_c + (1-u)v\mathbf{p}_d \, .$$

Les dérivées partielles de f sont données par

$$\begin{cases} \dfrac{\partial f}{\partial u}(u,v) = (1-v)(\mathbf{p}_b - \mathbf{p}_a) + v(\mathbf{p}_c - \mathbf{p}_d) \\ \\ \dfrac{\partial f}{\partial v}(u,v) = (1-u)(\mathbf{p}_d - \mathbf{p}_a) + u(\mathbf{p}_c - \mathbf{p}_b) \, . \end{cases}$$

La normale à la surface s'exprime ainsi également de manière bilinéaire

$$\begin{array}{rcrl} n(u,v) & = & \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v} & \\ \\ & = & (1-u)(1-v) & (\mathbf{p}_b - \mathbf{p}_a) \times (\mathbf{p}_d - \mathbf{p}_a) \\ & + & u(1-v) & (\mathbf{p}_b - \mathbf{p}_a) \times (\mathbf{p}_c - \mathbf{p}_b) \\ & + & (1-u)v & (\mathbf{p}_c - \mathbf{p}_d) \times (\mathbf{p}_d - \mathbf{p}_a) \\ & + & uv & (\mathbf{p}_c - \mathbf{p}_d) \times (\mathbf{p}_c - \mathbf{p}_b) \end{array}$$

Nous pouvons désormais utiliser la relation de l'eq. (1.21), avec une direction $\mathbf{e}$ quelconque donnant le volume associé au carreau.

Posons

$$\mathbf{q}_f = \begin{pmatrix} < \mathbf{p}_a, \mathbf{e} > \\ < \mathbf{p}_b, \mathbf{e} > \\ < \mathbf{p}_c, \mathbf{e} > \\ < \mathbf{p}_d, \mathbf{e} > \end{pmatrix} \quad \mathbf{q}_n = \begin{pmatrix} < (\mathbf{p}_b - \mathbf{p}_a) \times (\mathbf{p}_d - \mathbf{p}_a), \mathbf{e} > \\ < (\mathbf{p}_b - \mathbf{p}_a) \times (\mathbf{p}_c - \mathbf{p}_b), \mathbf{e} > \\ < (\mathbf{p}_c - \mathbf{p}_d) \times (\mathbf{p}_d - \mathbf{p}_a), \mathbf{e} > \\ < (\mathbf{p}_c - \mathbf{p}_d) \times (\mathbf{p}_c - \mathbf{p}_b), \mathbf{e} > \end{pmatrix} \, .$$

Le volume peut alors s'exprimer sous la forme matricielle :

$$V = \iint_{(u,v)\in[0,1]} \mathbf{q}_f^T \, \mathrm{m}(u,v) \, \mathbf{q}_n \, \mathrm{d}u \, \mathrm{d}v \, ,$$

avec

$$\mathrm{m}(u,v) = \begin{pmatrix} (1-u)^2(1-v)^2 & (1-u)u(1-v)^2 & (1-u)^2v(1-v) & (1-u)(1-v)uv \\ (1-v)^2(1-u)u & (1-v)^2u^2 & (1-v)vu(1-u) & (1-v)u^2v \\ v(1-v)(1-u)^2 & uv(1-u)(1-v) & v^2(1-u)^2 & v^2u(1-u) \\ uv(1-u)(1-v) & u^2v(1-v) & u(1-u)v^2 & u^2v^2 \end{pmatrix} .$$

En remarquant que seul m dépend de (u,v), on a

$$V = \mathbf{q}_f^T \left(\iint_{(u,v)\in[0,1]} \mathrm{m}(u,v) \mathrm{d}u \, \mathrm{d}v \right) \mathbf{q}_n \, ,$$

Notons $\mathrm{M} = \iint_{(u,v)\in[0,1]} \mathrm{m}(u,v) \mathrm{d}u \, \mathrm{d}v$, on a alors tout calculs faits

$$\mathrm{M} = \frac{1}{36} \begin{pmatrix} 4 & 2 & 2 & 1 \\ 2 & 4 & 1 & 2 \\ 2 & 1 & 4 & 2 \\ 1 & 2 & 2 & 4 \end{pmatrix} .$$

En considérant $\mathbf{e} = (0,0,1)$, on retombe sur l'expression donnée en eq. (2.1). Avec $\mathbf{q}_f = (z_a, z_b, z_c, z_d)$ et $\mathbf{q}_n$ revenant aux déterminants des coordonnées x et y.

ANNEXE

B

CALCUL DE LA CORRECTION DE VOLUME

Cette annexe démontre les différentes relations de correction de volume.

B.1 CORRECTION APPROCHÉE SUIVANT LES TROIS AXES (x, y, z)

Nous souhaitons résoudre le système de l'eq. (1.26) de la partie 1.4.2 et montrer que cette solution s'exprime par la relation. (1.28).

On rappel que $\mathbf{U} = (\mathbf{u}_1, \cdots, \mathbf{u}_{N_s}) \in \mathbb{R}^{3N_s}$ sont les vecteurs inconnus de déformation à appliquer. $\mathbf{U}_x \in \mathbb{R}^{N_s}$ représentant les composantes x de $\mathbf{U}$, et de manière similaire pour y et z.

La minimisation sous contrainte est la suivante :

$$\begin{cases} \min & \|\mathbf{U}_x\|^2 + \|\mathbf{U}_y\|^2 + \|\mathbf{U}_z\|^2 \\ \text{contraint à} & V(\mathbf{P}_1^x + \mathbf{U}_x, \mathbf{P}_1^Y + \mathbf{U}_y, \mathbf{P}_1^Z + \mathbf{U}_z) = \mathbf{V}(\mathbf{P}_0^x, \mathbf{P}_0^y, \mathbf{P}_0^z) \,, \end{cases}$$

avec $\mathbf{P}_0$ et $\mathbf{P}_1$ les vecteurs des positions de la surface originale et déformée par skinning. $\mathbf{P}^x$ représentant le vecteur des composantes x, et de même pour y et z.

L'expression du volume est cubique suivant les inconnues $\mathbf{U}_x, \mathbf{U}_y$ et $\mathbf{U}_z$. En linéarisant cette expression en considérant que les termes d'ordres supérieurs sont négligeables tel que décrit en eq. (1.27), la contrainte de volume peut alors s'ecrire

$$< \mathbf{U}_x, \nabla_{\mathbf{x}} V > + < \mathbf{U}_y, \nabla_{\mathbf{y}} V > + < \mathbf{U}_z, \nabla_{\mathbf{z}} V >= \Delta V \,.$$

L'équation est désormais linéaire suivant chacun des axes, nous provons traiter ceux-ci de manière indépendante. On considère désormais qu'un tiers de l'erreur en volume est corrigée

suivant chacun des axes (x, y, z). On peut alors scinder le système en 3 équations similaires :

$$\left\{ \begin{array}{ll} \min & \|\mathbf{U}_x\|^2 \\ \text{contraint à} & < \mathbf{U}_x, \nabla_{\mathbf{x}} V >= \Delta V/3 \, , \end{array} \right.$$

et de même avec y, et z.

Nous montrons dans la suite, la résolution de cette relation sur l'axe x.
L'utilisation des multiplicateurs de Lagrange permettent d'écrire l'energie non contrainte à minimiser associée

$$E(\mathbf{U}, \lambda) = \|\mathbf{U}_x\|^2 + \lambda \left(< \mathbf{U}_x, \nabla_{\mathbf{x}} V > -1/3 \Delta V \right) \, .$$

L'annulation des dérivées partielles [QSS00] de E par rapport à $\mathbf{U}_x$ et λ nous donne le système suivant à résoudre

$$\left\{ \begin{array}{l} 2\mathbf{U}_x + \lambda \nabla_{\mathbf{x}} V = 0 \\ < \mathbf{U}_x, \nabla_{\mathbf{x}} V >= \Delta V/3 \, . \end{array} \right.$$

Nous tirons de la première égalité $\mathbf{U}_x = -\frac{\lambda}{2} \nabla_{\mathbf{x}} V$. En injectant cette relation dans la seconde égalité, il vient

$$\lambda = -\frac{2 \Delta V}{3 \left(\nabla_{\mathbf{x}} V \right)^2} \, .$$

Finalement, nous obtenons donc par ré-injection dans la première relation

$$\mathbf{U}_x = \frac{\Delta V}{3} \frac{\nabla_{\mathbf{x}} V}{\left(\nabla_{\mathbf{x}} V \right)^2}$$

En agissant de manière identique suivant y et z, nous retrouvons la solution attendue

$$(\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z) = \frac{\Delta V}{3} \left(\frac{\nabla_{\mathbf{x}} V}{\left(\nabla_{\mathbf{x}} V \right)^2}, \frac{\nabla_{\mathbf{y}} V}{\left(\nabla_{\mathbf{y}} V \right)^2}, \frac{\nabla_{\mathbf{z}} V}{\left(\nabla_{\mathbf{z}} V \right)^2} \right) \, .$$

B.2 CORRECTION APPROCHÉE PONDÉRÉE SUIVANT LA DIRECTION NORMALE

Nous nous intéressons cette fois à l'eq. (2.3) en démontrant l'expression de sa solution donnée par l'eq. (2.4) en partie 2.3.1.

Soit la relation à minimiser

$$\left\{ \begin{array}{ll} \min & \displaystyle\sum_{k=0}^{N_s} \frac{\|\rho_k\|^2}{\gamma_k} \\ \\ \text{contraint à} & V((\mathbf{p_i} + \rho_i \mathbf{n}_i)_{i=[\![1,N_s]\!]}) = V_{\text{target}} \, . \end{array} \right.$$

En linéarisant l'expression du volume de la surface déformée suivant ρ_i, on a

$$V((\mathbf{p_i} + \rho_i \mathbf{n}_i)_{i=[\![1,N_s]\!]}) \simeq V((\mathbf{p_i})_{i=[\![1,N_s]\!]}) + \sum_{k=1}^{N_s} < \rho_k \mathbf{n}_k, \nabla V(\mathbf{p}_k) > \, .$$

La contrainte approximée s'exprime alors par

$$\sum_{k=1}^{N_s} < \rho_k \mathbf{n}_k, \nabla V(\mathbf{p}_k) >= \Delta V \ .$$

La fonctionnelle d'energie associé à l'utilisation des multiplicateurs de Lagrange est donnée par

$$E((\rho_i)_{i=[\![1,N_s]\!]}, \lambda) = \sum_{k=0}^{N_s} \frac{\|\rho_k\|^2}{\gamma_k} + \lambda \left(\sum_{k=1}^{N_s} < \rho_k \mathbf{n}_k, \nabla V(\mathbf{p}_k) > -\Delta V \right) \ .$$

L'annulation des dérivées partielles de E donne le système d'équation à résoudre

$$\begin{cases} \forall k \in [\![1, N_s]\!] \ , \quad 2\dfrac{\rho_k}{\gamma_k} + \lambda < \mathbf{n}_k, \nabla V(\mathbf{p}_k) >= 0 \\ \\ \displaystyle\sum_{k=1}^{N_s} < \rho_k \mathbf{n}_k, \nabla V(\mathbf{p}_k) >= \Delta V \end{cases}$$

En isolant ρ_k dans la première relation et en l'injectant dans la seconde, nous tirons

$$\lambda = -2 \frac{\Delta V}{\sum_{j=1}^{N_s} \gamma_j < \mathbf{n}_j, \nabla V(\mathbf{p}_j) >^2} \ .$$

En remplacant cette valeur de λ dans la première relation, nous obtenons finallement l'expression analytique attendue

$$\forall k \in [\![1, N_s]\!] \ , \quad \rho_k = \Delta V \frac{\gamma_k < \mathbf{n}_k, \nabla V(\mathbf{p}_k)}{\displaystyle\sum_{j=1}^{N_s} \gamma_j < \mathbf{n}_j, \nabla V(\mathbf{p}_j) >^2} \ .$$

B.3 CORRECTION APPROCHÉE PONDÉRÉE EN 3 ÉTAPES SUCCESSIVES

Nous démontrons ici le résultat de l'eq. (2.6).
Pour cela, nous procédons en trois étapes :

- Une première correction le long d'un axe x corrigeant le volume par un facteur μ_x.
- À partir de ce résultat intérmédiaire, nous appliquons une seconde correction le long de l'axe y corrigeant le volume par un facteur μ_y.
- Enfin, la dernière correction est appliquée le long de l'axe z corrigeant le dernier pourcentage de volume, soit $\mu_z = 1 - \mu_x - \mu_y$.

Résolution suivant l'axe x

La minimisation à résoudre suivant l'axe x de variable $(u_k^x)_{k\in[\![1,N_s]\!]}$ peut s'exprimer par

$$\begin{cases} \min & \displaystyle\sum_{k=1}^{N_s} \frac{\|u_k^x\|^2}{\gamma_k} \\ \\ \text{contraint à} & V(p_k^x + u_k^x, p_k^y, p_k^z) = \mu_x V_{\text{target}} + (1-\mu_x) V(p_k^x, p_k^y, p_k^z) \ . \end{cases}$$

En notant que la trilinéarité de l'expression du volume permet d'écrire

$$V(p_k^x + u_k^x, p_k^y, p_k^z) = V(p_k^x, p_k^y, p_k^z) + \sum_{k=1}^{N_s} < u_k^x, \nabla_{\mathbf{x}} V(\mathbf{p}_k) > \ ,$$

il vient la nouvelle expression de la contrainte donnant ce système à minimiser :

$$\begin{cases} \min & \displaystyle\sum_{k=1}^{N_s} \frac{\|u_k^x\|^2}{\gamma_k} \\ \text{contraint à} & \displaystyle\sum_{k=1}^{N_s} u_k^x \nabla_{\mathbf{x}} V(\mathbf{p}_k) = \mu_x \, \Delta V \ . \end{cases}$$

En utilisant la méthode des multiplicateurs de Lagrange, la fonctionnelle d'energie représentant l'équation non contrainte est donnée par

$$E((u_k^x)_{k\in[\![1,N_s]\!]}, \lambda) = \sum_{k=1}^{N_s} \frac{\|u_k^x\|^2}{\gamma_k} + \lambda \left(\sum_{k=1}^{N_s} u_k^x \nabla_{\mathbf{x}} V(\mathbf{p}_k) - \mu_x \, \Delta V \right)$$

L'annulation des dérivées partielles de E implique

$$\begin{cases} \forall k \in [\![1, N_s]\!] \ , \ \ 2\dfrac{u_k^x}{\gamma_k} + \lambda \nabla_{\mathbf{x}} V(\mathbf{p}_k) \\ \displaystyle\sum_{k=1}^{N_s} u_k^x \nabla_{\mathbf{x}} V(\mathbf{p}_k) = \mu_x \, \Delta V \end{cases}$$

De la première relation, nous tirons

$$u_k^x = -\frac{\lambda}{2} \gamma_k \nabla_{\mathbf{x}} V(\mathbf{p}_k)$$

Injectée dans la seconde égalitée, nous obtenons alors

$$\lambda = -2\mu_x \frac{\Delta V}{\sum_{k=1}^{N_s} \gamma_k \left(\nabla_{\mathbf{x}} V(\mathbf{p}_k)\right)^2}$$

Ce qui permet d'exprimer la valeur de u_k^x :

$$u_k^x = \mu_x \Delta V \ \frac{\gamma_k \nabla_{\mathbf{x}} V(\mathbf{p}_k)}{\sum_{j=1}^{N_s} \gamma_j \left(\nabla_{\mathbf{x}} V(\mathbf{p}_k)\right)^2}$$

En notant la norme modifiée $\|\mathbf{a}\|_{1/\gamma}^2 = \sum_{k=1}^{N_s} \gamma_k a_k^2$, nous pouvons alors écrire

$$\forall k \in [\![1, N_s]\!] \ , \ \ u_k^x = \mu_x \Delta V \ \frac{\gamma_k \nabla_{\mathbf{x}} V(\mathbf{p}_k)}{\|\nabla_{\mathbf{x}} V\|_{1/\gamma}^2}$$

Expression complète de la solution

Une fois cette solution trouvée, nous pouvons exprimer la nouvelle déformation à appliquer suivant y. Le nouveau système à minimiser est alors donné par

$$\begin{cases} \min & \displaystyle\sum_{k=1}^{N_s} \frac{\|u_k^y\|^2}{\gamma_k} \\ \text{contraint à} & V(p_k^x + u_k^x, p_k^y + u_k^y, p_k^z) = (\mu_x + \mu_y) V_{\text{target}} + (1 - (\mu_x + \mu_y)) V(p_k^x, p_k^y, p_k^z) \, . \end{cases}$$

Notons qu'ici u_k^x est désormais une variable connue, alors que u_k^y sont les inconnues.

En utilisant la trilinéarité de V dans la contrainte, on peut ré-écrire son expression en fonction du volume de $V(p_k^x + u_k^x, p_k^y, p_k^z)$. On a alors

$$\begin{cases} \min & \displaystyle\sum_{k=1}^{N_s} \frac{\|u_k^x\|^2}{\gamma_k} \\ \text{contraint à} & \displaystyle\sum_{k=1}^{N_s} u_k^y \nabla_{\mathbf{y}} V^\star(\mathbf{p}_k) = \mu_y \Delta V \, , \end{cases}$$

en dénotant le gradient de l'expression du volume issu de la surface déforme par u^x $\nabla_{\mathbf{x}} V^\star$, nous pouvons écrire directement la solution par analogie à l'étude précédente

$$\forall k \in [\![1, N_s]\!] \, , \quad u_k^x = \mu_x \Delta V \, \frac{\gamma_k \nabla_{\mathbf{x}} V^\star(\mathbf{p}_k)}{\|\nabla_{\mathbf{x}} V^\star\|_{1/\gamma}^2}$$

De même, on retrouvera pour z le gradient après déformation suivant x et y que nous nommons $\nabla_{\mathbf{z}} V^{\star\star}$.

Nous retrouvons ainsi l'expression analytique exacte attendue

$$\forall k \in [\![1, N_s]\!] \, , \quad \mathbf{u}_k = \gamma_k \, \Delta V \left(\mu_x \frac{\nabla_{\mathbf{x}} V(\mathbf{p}_k)}{\|\nabla_{\mathbf{x}} V\|_{1/\gamma}^2}, \mu_y \frac{\nabla_{\mathbf{y}} V^\star(\mathbf{p}_k)}{\|\nabla_{\mathbf{y}} V^\star\|_{1/\gamma}^2}, \mu_z \frac{\nabla_{\mathbf{z}} V^{\star\star}(\mathbf{p}_k)}{\|\nabla_{\mathbf{z}} V^{\star\star}\|_{1/\gamma}^2} \right)$$

B.4 CORRECTION APPROCHÉE ET PONDÉRÉE DE VOLUME ET D'AIRE SUIVANT LA DIRECTION NORMALE

Nous démontrons dans cette partie la relation (2.13) de la partie 2.8.

Considérons la minimisation à résoudre de l'eq. (2.12) en exprimant les contraintes après l'étape d'approximation linéaire :

$$\begin{cases} \min & \displaystyle\sum_{k=1}^{N_s} \frac{\rho_k^2}{\gamma_k} \\ \text{contraint à} & \displaystyle\sum_{k=1}^{N_s} \rho_k < \mathbf{n}_k, \nabla V(\mathbf{p}_k) >= \Delta V \\ \text{contraint à} & \displaystyle\sum_{k=1}^{N_s} \rho_k < \mathbf{n}_k, \nabla A(\mathbf{p}_k) >= \Delta A \, . \end{cases} \tag{B.1}$$

En suivant la méthode des multiplicateurs de Lagrange, l'expression de cette minimisation sous contrainte reviens à minimiser la fonctionnelle d'energie non contrainte

$$E((\rho_k)_{k\in[\![1,N_s]\!]},\lambda_1,\lambda_2)=\sum_{k=1}^{N_s}\frac{\rho_k^2}{\gamma_k} \quad +\lambda_1\left(\sum_{k=1}^{N_s}\rho_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)>-\Delta V\right) \\ +\lambda_2\left(\sum_{k=1}^{N_s}\rho_k<\mathbf{n}_k,\nabla A(\mathbf{p}_k)>-\Delta A\right)$$

L'annulation des dérivées partielles de E implique que

$$\forall k\in[\![1,N_s]\!]\ ,\ \ 2\frac{\rho_k}{\gamma_k}+\lambda_1<\mathbf{n}_k,\nabla V(\mathbf{p}_k)>+\lambda_2<\mathbf{n}_k,\nabla A(\mathbf{p}_k)>=0$$

En introduisant l'expression de ρ_k dans les deux contraintes, nous obtenons le système

$$\left\{\begin{array}{l}\lambda_1\displaystyle\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)>^2+\lambda_2\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)><\mathbf{n}_k,\nabla A(\mathbf{p}_k)>=-2\Delta V\\ \lambda_1\displaystyle\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)><\mathbf{n}_k,\nabla A(\mathbf{p}_k)>+\lambda_2\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)>^2=-2\Delta A\end{array}\right.$$

En introduisant les notations

$$\left\{\begin{array}{l}\|\mathbf{N}_V\|^2_{1/\gamma}=\displaystyle\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)>^2\\ \|\mathbf{N}_A\|^2_{1/\gamma}=\displaystyle\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla A(\mathbf{p}_k)>^2\\ <\mathbf{N}_V,\mathbf{N}_A>_{1/\gamma}=\displaystyle\sum_{k=1}^{N_s}\gamma_k<\mathbf{n}_k,\nabla V(\mathbf{p}_k)><\mathbf{n}_k,\nabla A(\mathbf{p}_k)>\end{array}\right.$$

on peut alors exprimer le système en (λ_1,λ_2) sous forme matricielle

$$\left(\begin{array}{cc}\|\mathbf{N}_V\|^2_{1/\gamma} & <\mathbf{N}_V,\mathbf{N}_A>_{1/\gamma}\\ <\mathbf{N}_V,\mathbf{N}_A>_{1/\gamma} & \|\mathbf{N}_A\|^2_{1/\gamma}\end{array}\right)\left(\begin{array}{c}\lambda_1\\ \lambda_2\end{array}\right)=-2\left(\begin{array}{c}\Delta V\\ \Delta A\end{array}\right)$$

La solution analytique est donc obtenue en ré-injectant la solution du système dans l'expression de ρ_k. On obtient alors bien la relation attendue :

$$\begin{array}{rcl}\forall k\in[\![1,N_s]\!]\ ,\ \ \rho_k & = & \dfrac{\gamma_k}{\|\mathbf{N}_V\|^2_{1/\gamma}\|\mathbf{N}_A\|^2_{1/\gamma}-<\mathbf{N}_V,\mathbf{N}_A>^2_{1/\gamma}}\\ & & \Big[\Big(\Delta V\|\mathbf{N}_V\|^2_{1/\gamma}-\Delta A<\mathbf{N}_V,\mathbf{N}_A>_{1/\gamma}\Big)<\mathbf{n},\nabla V(\mathbf{p}_k)>\\ & + & \Big(\Delta A\|\mathbf{N}_A\|^2_{1/\gamma}-\Delta V<\mathbf{N}_V,\mathbf{N}_A>_{1/\gamma}\Big)<\mathbf{n},\nabla A(\mathbf{p}_k)>\Big]\ .\end{array}$$

ANNEXE

C

RELATION ENTRE COMPRESSION ET PROFONDEUR DU PLIS

C.1 RELATION ENTRE COMPRESSION ET PROFONDEUR DU PLIS

Nous démontrons ici la relation de l'eq. (3.9) en partie 3.3.1.

Soit $R(u)$ le rayon de courbure du plis situé sur la courbe paramétrée par u. Dans la suite, nous nous placons à u constant. Nous omettons de mentionner ce paramètre pour simplifier la notation.

Faisons la supposition que la compression v soit constante sur la largeur apparente du plis que l'on notera $2r$ (voir fig. 3.15), avec

$$r^2 = R^2 - \beta^2 \, .$$

Cette région de largeur $2r$ doit compenser la compression de $(1-v)$. La longueur curviligne du plis généré L doit ainsi être égale à

$$L = 2r/(1-v) \, ,$$

afin de compenser cette compression.

Notons également que d'après les critères géométriques et de trigonométrie sur la fig. 3.15, l'arc curviligne possède une longueur

$$L = 2R \operatorname{acos}\left(\frac{\beta}{R}\right) \, .$$

En égalisant les deux expressions de L, nous obtenons donc la relation à vérifier

$$(1-v)\operatorname{acos}\left(\frac{\beta}{R}\right) - \sqrt{1-\left(\frac{\beta}{R}\right)} = 0 \, .$$

C.2 PONDÉRATION DU NOYAUX DE CAUCHY À RAYON FIXÉ.

Nous démontrons ici la relation de l'eq. (3.10). Considérons l'intégrale pondérée du noyaux de cauchy

$$F(\mathbf{p}) = \int_u \frac{\omega(u)}{\left(1 + s^2 \|c(u) - \mathbf{p}\|\right)^2} \, \mathrm{d}u \, .$$

Supposons que ω soit contant, et que la courbe soit un segment de longueur infinie par rapport à la distance d'évaluation du potentiel en $\mathbf{p}$. Considérons que cette courbe soit confondu avec l'axe $x = 0$. On a alors

$$F(\mathbf{p}) = \int_{-\infty}^{+\infty} \omega \frac{\mathrm{d}x}{\left(1 + s^2(x^2 + \|\mathbf{p}\|^2)\right)^2} \, .$$

Le résultat de cette intégrale sur $\mathbb{R}$ est connue [BSMM04] et vaut

$$F(\mathbf{p}) = \frac{\omega\pi}{2s(1 + s^2\|\mathbf{p}\|^2)^{3/2}} \, .$$

Nous souhaitons désormais que pour $\|\mathbf{p}_0\| = R$, nous ayant $F(\mathbf{p}_0) = 1$. Nous tirons donc la relation attendue

$$\omega = \frac{2s}{\pi} \left(1 + s^2 R^2\right)^{3/2} \, .$$

ANNEXE

D

ECHANTILLONNAGE RÉGULIER SUIVANT LA LONGUEUR CURVILIGNE D'UNE COURBE CUBIQUE

Longueur curviligne relative

Soit une courbe cubique (notons que la démarche présentée se généralise trivialement à des polynome de degré quelconque) c telle que

$$c : \left\{ \begin{array}{ccc} [0,1] & \rightarrow & \mathbb{R}^3 \\ t & \mapsto & c(t) = At^3 + Bt^2 + Ct + D \, . \end{array} \right.$$

Soit L_c la longueur de cette courbe. On cherche le paramètre t_0 tel que l'abscisse curviligne relative soit donnée par la valeur souhaitée u_0. C'est à dire que

$$u_0 = \frac{L(t_0)}{L_c} \, ,$$

avec $L(t)$ la longueur curviligne estimée au paramètre t. On a donc formellement

$$L(t) = \int_{x=0}^{t} \| c'(x) \| \, \mathrm{d}x \, .$$

L n'étant pas exprimable analytiquement, nous reformulons le problème sous forme de la recherche d'une racine d'une fonction non linéaire. Trouver t_0 étant donné la valeur de l'abscisse curviligne relative u_0 revient à trouver les zéros de la fonction F telle que

$$F(t) = \frac{1}{L_c} \int_{0}^{t} \| c'(x) \| \, \mathrm{d}x \ - u_0 \, .$$

Racine de l'équation

Pour chercher les zéros de cette fonction, nous utilisons la méthode de recherche de racines de Newton [QSS00]. Nous savons qu'il n'existe qu'une unique solution à l'équation $F(t_0) = 0$ et nous savons que l'on a $t_0 \simeq u_0$ pour une courbe cubique.

L'algorithme consiste donc à itérer sur la valeur de t que l'on dénotera t^n à l'itération n en suivant

$$\left\{ \begin{array}{lcl} t^0 & = & u_0 \\ t^{n+1} & = & t^n - \frac{F(t^n)}{F'(t^n)} \end{array} \right. ,$$

jusqu'à convergence, avec $F'(t) = \|c'(t)\|/L_c$.

Intégration numérique

On notera que, pour t donné, l'expression de F fait intervenir une intégrale non exprimable analytiquement. Nous approximons celle-ci par une quadrature de Gauss-Legendre [QSS00]. C'est à dire que l'on considère

$$\int_0^t \|c'(x)\| \mathrm{d}x \simeq \frac{t}{2} \sum_{k=0}^{N_g} \omega_k \left\| c' \left(\frac{x_k + 1}{2} t \right) \right\| ,$$

où les x_k sont les racines des polynomes de Legendre entre $[-1, 1]$, et les ω_k les poids de pondérations associés. Ces valeurs sont tabulées dans la littérature.

En pratique, on utilisera $N_g = 8$ donnant une très bonne approximation de la longueur de la courbe. L'évaluation de F reviens donc à 8 évaluations de la norme de la dérivée de c.

Algorithme d'évaluation

Nous avons donc détaillé une méthode d'échantillonnage de courbe cubique permettant de repartir ceux-ci en fonction de leur abscisse curviligne.

L'algorithme suivit est le suivant

```
function echantillonnage(c,x)
 t0=x
 Tant que non convergence
   F=quadrature(c,t0)
   F'=||c'(t0)||
   t0=t0-F/F'
 Fin tant que
```

Un exemple de résultat est montré en fig. D.1.

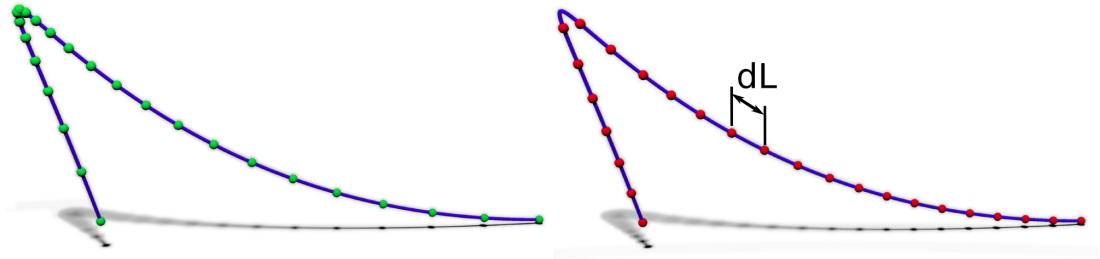


FIGURE D.1 – *Échantillonnage d'une courbe cubique. La courbe de gauche correspond à un échantillonnage uniforme dans l'espace des paramètres de la cubique alors que la figure de droite est échantillonnée régulièrement suivant la distance curviligne. Dans le cas d'une correspondance avec un segment de droite isométrique à la cubique, les sommets seront régulièrement espacés sur le segment également.*

INDEX

A

Anisotropie 115
Apprentissage 63
Arêtes 81, 186

C

Cage 47
Carte de normales 87
Champ de vecteurs 48, 71
Collision 128, 160
Connectivité 28
Contrainte
  Aire 136
  Longueur 80, 154
  Volume 70, 107
Coordonnée
  Cartésienne 23
  Laplacienne 68, 75
Courbe
  Bézier,de 46
  Cubique 181, 213
  Hermite, de 182
Courbe de pli 88, 148

D

Déformation
  Espace, de l' 44
  Exemple, basée 55, 62
  Physique 37, 84
  Surface, de 24, 43, 155
  Surfaces, basée 65
Déformeur implicite 155
Delaunay, triangulation 158, 181, 186
Dual Quaternion 57

F

Free Form Deformation 46, 74

G

Géodésique, distance 60, 97

H

Hand-Navigator 131

I

Isométrie 81

L

Ligne de champ 148

M

Maillage 25, 28
  Étoilé 83

Mapping 23
Masses ressorts 38, 162, 186
Multiplicateur de Lagrange 77, 84

O

Origami 90

P

Paramétrisation 145
Poids de skinning 59, 95
Polygone 28
Poses clés 62

Q

Quadrangle 29, 100

R

Rayon de courbure 154

S

Skinning 50, 94, 163
Spline 27
Squelette d'animation 50, 133
Subdivision locale 127, 158
Surface 21
    Bilinéaire 100, 203
    Développable 82, 89, 143, 171
    Implicite 80, 155
    Subdivision 66
Sweepers 48

T

Tenseur d'élongation 146
Topologie 31

V

Variété 23
Volume, changement 99

W

Wires 50

BIBLIOGRAPHIE

[ABCO+01] Marc Alexa, Johannes Behr, Daniel Corhen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva, *Point set surfaces*, Proceedings of the conference on Visualization (2001).

[ACWK04a] Alexis Angelidis, Marie-Paule Cani, Geoff Wyvill, and Scott King, *Swirling-Sweepers : Constant Volume Modeling*, Sketches ACM SIGGRAPH (2004).

[ACWK04b] Alexis Angelidis, Marie-Paule Cani, Geoff Wyvill, and Scott King, *Swirling-Sweepers : Constant Volume Modeling*, Graphical Models (GMOD). Proceedings of Pacific Graphics **68** (2004), no. 4.

[AHLD07] Grégoire Aujay, Franck Hétroy, Francis Lazarus, and Christine Depraz, *Harmonic skeleton for realistic character animation*, Eurographics / ACM SIGGRAPH Symposium on Computer Animation (SCA) (2007).

[Ale02] Marc Alexa, *Linear combination of transformations*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH **21** (2002), no. 3.

[AN06] Mikko Alava and Kaarlo Niskanen, *The physics of paper*, Reports on Progress in Physics **69** (2006), 669–723.

[ANCO05] Marc Alexa Andrew Nealen, Olga Sorkine and Daniel Cohen-Or, *A sketch-based interface for detail-preserving mesh editing*, ACM Transaction on Graphics (TOG). Proccedings of ACM SIGGRAPH **24** (2005), no. 3, 1142–1147.

[AS06] Alexis Angelidis and Karan Singh, *Space deformations and their application to shape modeling*, ACM SIGGRAPH, Course Notes (2006).

[AS07] Alexis Angelidis and Karan Singh, *Kinodynamic skinning using volume-preserving deformations.*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) (2007).

[Aum03] Günter Aumann, *A simple algorithm for designing developable bézier surfaces*, Computer Aided Geometric Design (CAGD) (2003), 601–619.

[AWC04] Alexis Angelidis, Geoff Wyvill, and Marie-Paule Cani, *Sweepers : Swept User-Defined Tools for Modeling by Deformation*, Shape Modeling Applications (SMI) (2004), 63–73.

[Bar84] Alan Barr, *Global and Local Deformations of Solid Primitives*, ACM SIGGRAPH **18** (1984), no. 3.

[BB08] Jan Bender and Daniel Bayer, *Parallel simulation of inextensible cloth*, Virtual Reality Interactions and Physical Simulations (VRIPHYS) (2008).

[BCWG09] Mirela Ben-Chen, Ofir Weber, and Craig Gotsman, *Variational harmonic maps for space deformation*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH **28** (2009), no. 3.

[BFA02] Robert Bridson, Ronald Fedkiw, and John Anderson, *Robust treatment of collisions, contact and friction for cloth animation*, ACM SIGGRAPH, Computer Graphics (2002).

[BFM+04] David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, and Ross Whitaker, *Level set and PDE methods for computer graphics*, ACM SIGGRAPH, Course Notes (2004).

[BGW06] Robert Burgoon, Eitan Grinspun, and Zoe Wood, *Discrete shells origami*, Proc. of Computers and their Applications. (2006), 180–187.

[BHW94] David E. Breen, Donald H. House, and Michael J. Wozny, *Predicting the drape of woven cloth using interacting particles*, ACM SIGGRAPH. Computer Graphics (1994), 365–372.

[BKL+09] Mario Botsh, Leif Kobbelt, Bruno Levy, Pierre Alliez, and Mark Pauly, *Polygon mesh processing (A K Peters)*, 2009.

[Ble] Blender, *http ://www.blender.org/*.

[Bli82] James F. Blinn, *A generalization of algebraic surface drawing*, ACM SIGGRAPH Computer Graphics **16** (1982), no. 3.

[Blo97] Jules Bloomenthal, *Bulge elimination in convolution surfaces*, Computer Graphics Forum (CGF) **16** (1997), no. 1, 31–41.

[BMF03] Robert Bridson, Sebastian Marino, and Ronald Fedkiw, *Simulation of clothing with folds and wrinkles*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) (2003).

[BMWG07] Miklós Bergou, Saurabh Mathur, Max Wardetzky, and Eitan Grinspun, *TRACKS : Toward directable thin shells*, ACM Transaction on Graphics (TOG), Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[BP07] Ilya Baran and Jovan Popovic, *Automatic rigging and animation of 3D characters*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[BPK+07] Mario Botsh, Mark Pauly, Leif Kobbelt, Pierre Alliez, Bruno Lévy, Stephan Bischoff, and Christian Rössl, *Geometric modeling based on polygonal meshes*, ACM SIGGRAPH Course Notes (2007).

[BS91] Jules Bloomenthal and Ken Shoemake, *Convolution surfaces*, Computer Graphics **25** (1991), no. 4, 251–256.

[BSMM04] I. N. Bronshtein, K. A. Semendyayev, G. Musiol, and H. Muehlig, *Handbook of mathematics*, Springer, 2004.

[BW98] David Baraff and Andrew Witkin, *Large steps in cloth simulation*, ACM SIGGRAPH, Computer Graphics (1998).

[BW07] Pengbo Bo and Wenping Wang, *Geodesic-controlled developable surfaces for modeling paper bending*, Computer Graphics Forum. **26** (2007).

[CAL+10] Jean-Rémy Chardonnet, André De Carvalho Amaro, Jean-Claude Léon, Damien Huyghe, and Marie-Paule Cani, *Designing and evolving hands-on interaction prototypes for virtual reality*, Virtual Reality International Conference (VRIC) (2010), 25–34.

[CALC09a] Jean-Rémy Chardonnet, André De Carvalho Amaro, Jean-Claude Léon, and Marie-Paule Cani, *Hand navigator : Prototypages de périphériques d'interaction pour le contrôle d'une main virtuelle*, Journées de l'Association Française de Réalité Virtuelle (AFRV) (2009).

[CALC09b] Jean-Rémy Chardonnet, André De Carvalho Amaro, Jean-Claude Léon, and Marie-Paule Cani, *Hand Navigator : Experimenting hand navigation in desktop virtual reality - Demo paper*, EGVE/ICAT/EuroVR Joint Virtual Reality Conference (2009).

[Car76] Manfredo Do Carmo, *Differential geometry of curves and surfaces*, Prentice Hall, 1976.

[CC78] E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer Aided Design (CAD) **10** (1978), no. 6, 350–355.

[CD97] Marie-Paule Cani and Mathieu Desbrun, *Animation of Deformable Models Using Implicit Surfaces*, IEEE Transaction on Visualization and Computer Graphics (TVCG) **3** (1997), no. 1, 39–50.

[CGA] CGAL, *http ://www.cgal.org/*.

[CGW+05] Lawrence D. Cutler, Reid Gershbein, Xiaohuan C. Wang, Cassidy Curtis, Erwan Maigret, Luca Prasso, and Peter Farson, *An art-directed wrinkle system for CG character clothing*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) (2005).

[ChZK98] Yan Chen, Qing hong Zhu, and Arie Kaufman, *Physically-based animation of volumetric objects*, Proceedings of the Computer Animation (1998).

[CK02] Kwang-Jin Choi and Hyeong-Seok Ko, *Stable but responsive cloth*, ACM Transactions on Graphics. **21** (2002), 604–611.

[CK05] Kwang-Jin Choi and Hyeong-Seok Ko, *Research problems in clothing simulation*, Computer Aided Design (CAD) **37** (2005), 585–592.

[CM98] J. Chalfant and T. Maekawa, *Design for manufacturing using B-spline developable surfaces*, Journal of Ship Research (1998).

[Coq90] Sabine Coquillart, *Extended Free-Form Deformation : A Sculpturing Tool for 3D Geometric Modeling*, ACM SIGGRAPH **24** (1990), no. 4.

[CS02] Chih-Hsing Chua and Carlo H. Séquin, *Developable Bézier patches : properties and design*, Computer Aided Design (CAD) **34** (2002), no. 7, 511–527.

[CT10] Ming Chen and Kai Tang, *A fully geometric approach for developable cloth deformation simulation*, The Visual Computer. (2010).

[CYTT92] Michel Carignan, Ying Yang, Nadia Magnenat Thalmann, and Daniel Thalmann, *Dressing animated synthetic actors with complex deformable clothes*, ACM SIGGRAPH, Computer Graphics **26** (1992), no. 2.

[dASTH10] Edilson de Aguiar, Leonid Sigal, Adrien Treuille, and Jessica K. Hodgins, *Stable spaces for real-time clothing*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **29** (2010), no. 3.

[DC95] Mathieu Desbrun and Marie-Paule Cani, *Animating soft substances with implicit surfaces*, ACM SIGGRAPH, Computer Graphics (1995).

[DJW+06] Philippe Decaudin, Dan Julius, Jamie Wither, Laurence Boissieux, Alla Sheffer, and Marie-Paule Cani, *Virtual garments : A fully geometric approach for clothing design*, Computer Graphics Forum (CGF). Proccedings of Eurographics **25** (2006), no. 3, 625–634.

[DKT98] Tony DeRose, Michael Kass, and Tien Truong, *Subdivision surfaces in character animation*, ACM SIGGRAPH, Computer Graphics (1998).

[EB08] Elliot English and Robert Bridson, *Animating developable surfaces using nonconforming elements*, ACM Transactions on Graphics (TOG). Proceddings of ACM SIGGRAPH **27** (2008), 1–5.

[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** (2002), no. 1, 83–116.

[EKS03] Olaf Etzmuss, Michael Keckeisen, and Wolfgang Strasser, *A fast finite element solution for cloth modelling*, Proceedings of Pacific Graphics (PG) (2003).

[Elb00] Gershon Elber, *Linearizing the area and volume constraints*, Tech. report, Technion, CIS-2000-04, 2000.

[EMF02] Douglas Enright, Stephen Marschner, and Ronald Fedkiw, *Animation and rendering of complex water surfaces*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **21** (2002), 736–744.

[Far01] Gerald Farin, *Curves and surfaces for CAGD : A practical guide*, 2001.

[FB88] David Forsey and Richard Bartels, *Hierarchical B-Spline Refinement*, ACM SIGGRAPH **22** (1988), no. 4.

[FF01] Nick Foster and Ronald Fedkiw, *Practical animation of liquids*, ACM SIGGRAPH, Computer Graphics (2001), 15–22.

[FKR05] Michael S. Floater, Géza Kós, and Martin Reimers, *Mean value coordinates in 3D*, Computer Aided Geometric Design **22** (2005), no. 7, 623–631.

[For03] J. Fordham, *Middle earth strikes back*, Cinefex **92** (2003), 71–142.

[Fre02] William H. Frey, *Boundary triangulations approximating developable surfaces that interpolate a close space curve*, International Journal of Foundations of Computer Science (IJFCS) (2002).

[Fre04] William H. Frey, *Modeling buckled developable surfaces by triangulation*, Computer Aided Design. **36** (2004).

[FSJ01] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen, *Visual simulation of smoke*, ACM SIGGRAPH, Computer Graphics **23–30** (2001).

[GB08] James Gain and Dominique Bechmann, *A survey of spatial deformation from a user-centered perspective*, ACM Transaction on Graphics (TOG) **27** (2008), no. 4.

[GBFP11a] Benjamin Gilles, Guillaume Bousquet, Francois Faure, and Dinesh K. Pai, *Frame-based elastic models*, ACM Transaction on Graphics (TOG) (2011).

[GBFP11b] Benjamin Gilles, Guillaume Bousquet, Francois Faure, and Dinesh K. Pai, *Sparse meshless models of complex deformable objects*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH (2011).

[GG07] Gaël Guennebaud and Markus Gross, *Algebraic point set surfaces*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[GH95] André Guéziec and Robert Hummel, *Exploiting triangulated surface extraction using tetrahedral decomposition.*, IEEE Transactions on Visualization and Computer Graphics (TVCG) **1** (1995), no. 4, 328–342.

[GHDS03] Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder, *Discrete shells*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) (2003).

[GHF+07] Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun, *Efficient simulation of inextensible cloth*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[Gib99] Sarah F. Gibson, *Calculating the distance map for binary sampled data*, Tech. report, 1999.

[GIF09] GIF, *Grenoble inovation fair*, 2009.

[GOCP97] Carlos Gonzalez-Ochoa, Scott Mc Cammon, and Jörg Peters, *Computing moments of piecewise polynomial surfaces*, Technical Report TR-97-009, Purdue University, Indiana USA (1997).

[GOP99] Carlos Gonzalez-Ochoa and Jorg Peters, *Localized-hierarchy surface splines (LeSS)*, Symposium on Interactive 3D Graphics (I3D) (1999).

[Gra92] Jens Gravesen, *Adaptive subdivision and the length of Bezier curves*, Mat-report n. 1992-10, Mathematical Institute. Technical University of Denmark (1992).

[GWDS08] Eitan Grinspun, Max Wardetzky, Mathieu Desbrun, and Peter Schröder, *Discrete differential geometry : An applied introduction*, ACM SIGGRAPH Asia Course Notes (2008).

[Hau04] Michael Hauth, *Visual simulation of deformable models*, PhD Thesis, Tübingen (2004).

[HBVMT99] Sunil Hadap, Endre Bangarter, Pascal Volino, and Nadia Magnenat-Thalmann, *Animating wrinkles on clothes*, IEEE Visualization (1999).

[HGLT09] Franck Hétroy, Cédric Gérot, Lin Lu, and Boris Thibert, *Simple flexible skinning based on manifold modeling*, Conference on Computer Graphics Theory and Applications (GRAPP) (2009), 256–265.

[HLS07] Kai Horman, Bruno Lévy, and Alla Sheffer, *Mesh parameterization : Theory and practice*, ACM SIGGRAPH Course Notes (2007).

[HML99] Gentaro Hirota, Renee Maheshwari, and Ming C. Lin, *Fast volume-preserving free form deformation using multi-level optimization*, Symposium on Solid Modeling and Applications (1999), 234–245.

[HNW00] Ernst Hairer, Syvert P. Norsett, and Gerhard Wanner, *Solving ordinary differential equations i*, ch. II, Springer, 2000.

[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 Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **25** (2006), no. 3.

[JBPS11] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine, *Bounded biharmonic weights for real-time deformation*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH (2011).

[JK10] Henrik Jönsson and Pawel Krupinski, *Modeling plant growth and pattern formation*, Current Opinion in Plant Biology **13** (2010), no. 13, 5–11.

[JMD+07] Pushkar Joshi, Mark Meyer, Tony DeRose, Brian Green, and Tom Sanocki, *Harmonic coordinates for character articulation*, ACM Transactions on Graphics (TOG). Proccedings of ACM SIGGRAPH **26** (2007), no. 3.

[JSW05] Tao Ju, Scott Schaefer, and Joe Warren, *Mean Value Coordinates for Closed Triangular Meshes*, ACM Transaction on Graphics (TOG), Proceedings of ACM SIGGRAPH **24** (2005), no. 3.

[JT05] Doug L. James and Christopher D. Twigg, *Skinning mesh animations*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH **24** (2005), no. 3.

[Kaw80] Sueo Kawabata, *The standardization and analysis of hand evaluation*, The Textile Machinery Society of Japan (1980).

[KCBZ05] Hyeong-Seok Ko, Kwang-Jin Choi, Robert Bridson, and Dongliang Zhang, *Advanced topics on clothing simulation and animation*, ACM SIGGRAPH, Course Notes (2005).

[KCCL01] Young-Min Kang, Jeong-Hyeon Choi, Hwan-Gue Cho, and Do-Hoon Lee, *An efficient animation of wrinkled cloth with approximate implicit integration*, The Visual Computer **17** (2001), 147–157.

[KCZO07] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O'Sullivan, *Skinning with dual quaternions*, Symposium on Interactive 3D Graphics and Games (i3D) (2007).

[KCZO08] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O'Sullivan, *Geometric skinning with approximate dual quaternion blending*, ACM Transaction on Graphics (TOG) **27** (2008), no. 4.

[KJM08] Jonathan M. Kaldor, Doug L. James, and Steve Marschner, *Simulating knitted cloth at the yarn level*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **27** (2008), no. 3.

[KJP02] Paul G. Kry, Doug L. James, and Dinesh K. Pai, *Eigenskin : Real time large deformation character skinning in hardware*, ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA) (2002).

[KMB+09] Peter Kaufmann, Sebastian Martin, Mario Botsch, Eitan Grinspun, and Markus Gross, *Enrichment textures for detailed cutting of shells*, ACM Transactions on Graphics (TOG), Proceedings of ACM SIGGRAPH **28** (2009), no. 3.

[KMD+07] Ladislav Kavan, Rachel McDonnell, Simon Dobbyn, Jiri Zara, and Carol O'Sullivan, *Skinning arbitrary deformations*, Symposium on Interactive 3D Graphics and Games (i3D) (2007).

[KMP07] Martin Killian, Niloy J. Mitra, and Helmut Pottmann, *Geometric modeling in shape space*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[Kre91] Erwin Kreyszig, *Differential geometry*, Dover, 1991.

[Kro86] B. Kroyer, *Animating with a hierarchy*, ACM SIGGRAPH Course Notes (1986).

[KSO10] Ladislav Kavan, Peter-Pike Sloan, and Carol O'Sullivan, *Fast and efficient skinning of animated meshes*, Computer Graphics Forum (CGF). Proceedings of Eurographics **29** (2010), no. 2.

[KWH04] Stefan Kimmerle, Markus Wacker, and C. Holzer, *Multilayered wrinkle textures from strain*, Vision, Modeling and Visualization (VMV) (2004), 225–232.

[KZ05] Ladislav Kavan and Jiri Zara, *Spherical blend skinning : A real-time deformation of articulated models*, Symposium on Interactive 3D Graphics and Games (2005).

[KZC09] Y. M. Kang, H.-G. Zhang, and H.-G. Cho, *Plausible virtual paper for real time applications*, Computer Animation and Social Agents (CASA) - Short paper (2009).

[LA10] J. P. Lewis and K. Anjyo, *Direct manipulation blendshapes*, IEEE Computer Graphics and Application (CG&A) **30** (2010), no. 4, 42–50.

[Lan98] Jeff Lander, *Skin them bones : Game programming for the web generation*, Game Developer Magazine (1998), 11–16.

[Lan99] Jeff Lander, *Over my dead, polygonal body*, Game Developer Magazine (1999), 1–4.

[Las87] John Lasseter, *Principles of traditional animation applied to 3D computer animation*, ACM SIGGRAPH **21** (1987), no. 4, 35–44.

[LC87] William E. Lorensen and Harvey E. Cline, *Marching cubes : A high resolution 3D surface construction algorithm.*, ACM SIGGRAPH (1987).

[LC04] Caroline Larboulette and Marie-Paule Cani, *Real-time dynamic wrinkles*, Computer Graphics International (2004).

[LCF00] J. P. Lewis, Matt Cordner, and Nickson Fong, *Pose space deformation : A unified approach to shape interpolation and skeleton-driven deformation*, SIGGRAPH (2000).

[LCORL07] Yaron Lipman, Daniel Cohen-Or, Gal Ran, and David Levin, *Volume and shape preservation via moving frame manipulation*, ACM Transaction on Graphics (TOG) **26** (2007), no. 1.

[Lie94] Pascal Lienhardt, *N-Dimensional generalized combinatorial maps and cellular quasi-manifolds*, International Journal of Computational Geometry and Applications (IJCGA) **4** (1994), no. 4, 275–324.

[LKCOL07] Yaron Lipman, Johannes Kopf, Daniel Cohen-Or, and David Levin, *GPU-assisted positive mean value coordinate for mesh deformations*, Eurographics / ACM SIGGRAPH Symposium on Geometry Processing (SGP) (2007).

[LLCO08] Yaron Lipman, David Levin, and Daniel Cohen-Or, *Green coordinates*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH **27** (2008), no. 3.

[Loo87] Charles T. Loop, *Smooth subdivision surfaces based on triangles*, Master's thesis, University of Utah, 1987.

[LPW+06] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang, *Geometric modeling with conical meshes and developable surfaces*, ACM Transaction on Graphics. **25** (2006).

[LS10] Eric Landreneau and Scott Schaefer, *Scales and scale-like structures*, Computer Graphics Forum (CGF), Proceedings of Symposium on Geometry Processing (SGP) **29** (2010), no. 5, 1653–1659.

[LSCO+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 (SMI) (2004).

[LSLCO05] Yaron Lipman, Olga Sorkine, David Levin, and Daniel Cohen-Or, *Linear rotation-invariant coordinates for meshes*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **24** (2005), no. 3.

[lTZkF04] Chiew lan Tai, Hongxin Zhang, and Jacky Chun kin Fong, *Prototype modeling from sketched silhouettes based on convolution surfaces*, Computer Graphics Forum (CGF) **23** (2004), no. 1, 71–83.

[May] Maya, *http ://usa.autodesk.com/maya/*.

[MC10] Matthias Müller and Nuttapong Chentanez, *Wrinkle meshes*, Eurographics / ACM SIGGRAPH Symposium on Computer Animation (SCA) (2010).

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

[MG03] Alex Mohr and Michael Gleicher, *Building efficient, accurate character skins from examples*, ACM SIGGRRAPH (2003).

[Mil88] Gavin S. P. Miller, *The motion dynamics of snakes and worms*, ACM SIGGRAPH, Computer Graphics **22** (1988), 4.

[MS98] Jon McCormack and Andrei Sherstyuk, *Creating and rendering convolution surfaces*, Computer Graphics Forum (CGF) **17** (1998), no. 2, 113–120.

[MTCSP04] Nadia Magnenat-Thalmann, Frederic Cordier, Hyewon Seo, and George Papagianakis, *Modeling of bodies and clothes for virtual environments*, International Conference on Cyberworlds (CW) (2004).

[MTLT88] Nadia Magnenat-Thalmann, Richard Laperrière, and Daniel Thalmann, *Joint-dependent local deformations for hand animation and object grasping*, Proceedings on Graphics Interface (1988).

[Mur91] Shigeru Muraki, *Volumetric shape description of range data using blobby model*, ACM SIGGRAPH (1991).

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

[NFJ02] Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen, *Physically based modeling and animation of fire*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **21** (2002), 736–744.

[NHK+89] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura, *Object modeling by distribution function and a method of image generation*, The transactions of the Institute of Electronics and Communication Engineers of Japan **J68-D** (1985 (English translation by Takao Fujuwara, Advanced Studies in Computer Aided Art and Design, England 1989)), no. 4, 718–725.

[NMK+05] Andrew Nealen, Matthias Müller, Richard Keiser, Eddv Bowerman, and Mark Carlson, *Physically based deformable models in computer graphics*, Eurographics, State of The Art Report (2005).

[OF03] Stanley Osher and Ronald Fedkiw, *Level set methods and dynamic implicit surfaces*, Springer, 2003.

[Ope] OpenMesh, *http ://www.openmesh.org/*.

[OS88] Stanley Osher and James A. Sethian, *Fronts propagating with curvature-dependent speed : Algorithms based on hamilton-jacobi formulations*, Journal of Computational Physics **79** (1988), no. 1, 12–49.

[Par72] Frederick I. Parke, *Computer generated animation of faces*, ACM National Conference **1** (1972), 454–457.

[PB81] Stephen M. Platt and Norman I. Badler, *Animating facial expressions*, ACM SIGGRAPH, Computer Graphics **15** (1981), no. 3.

[Per08] Mathieu Perriollat, *Paramétrisation et reconstruction des surfaces développables à partir d'images*, Ph.D. thesis, Université Clermont, Blaise Pascal, 2008.

[Pet04] Martin Peternell, *Developable surface fitting to point clouds*, Computer Aided Geometric Design (CAGD) **21** (2004).

[PF95] Helmut Pottmann and Gerald Farin, *Developable rational Bézier and B-spline surface*, Computer Aided Geometric Design (CAGD) **12** (1995).

[PFA06] Xavier Pennec, Pierre Fillard, and Nicholas Ayache, *A riemannian framework for tensor computing*, Internation Journal of Computer Vision **66** (2006), no. 1, 41–66.

[PKST08] Simon Pabst, Sybille Krzywinski, Andrea Schenk, and Bernhard Thomaszewski, *Seams and Bending in Cloth Simulation*, Virtual Reality Interactions and Physical Simulations (VRIPHYS) (2008), 31–38.

[PO09] Eric G. Parker and James O'Brien, *Real-time deformation and fracture in a game environment*, ACM SIGGRAPH/Eurographics, Symposium on Computer Animation (SCA) (2009).

[PP93] Ulrich Pinkall and Konrad Polthier, *Computing discrete minimal surfaces and their conjugates*, Experiment. Math. **2** (1993), no. 1, 15–36.

[PR] POV-Ray, *http ://www.povray.org/*.

[Pro97] Xavier Provot, *Deformation constraints in a mass-spring model to describe rigid cloth behavior*, Graphics Interface. (1997), 147–154.

[PS07] F. Pérez and J. Suárez, *Quasi-developable B-spline surfaces in ship hull design*, Computer Aided Design (CAD) **39** (2007), 853–862.

[PZB+09] Tiberiu Popa, Qingnan Zhou, Derek Bradley, Vladislav Kraevoy, Hongbo Fu, Alla Sheffer, and Wolfgang Heidrich, *Wrinkling captured garments using space-time data-driven deformation*, Computer Graphics Forum. Proceedings of Eurographics **28** (2009).

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

[RCHT10] Damien Rohmer, Marie-Paule Cani, Stefanie Hahmann, and Boris Thibert, *Generation de surfaces isometriques a un patron par decoupage recursif*, Groupe de Travail en Modelisation Geometrique (GTMG) (2010).

[RCHT11] Damien Rohmer, Marie-Paule Cani, Stefanie Hahmann, and Boris Thibert, *Folded Paper Geometry from 2D Pattern and 3D Contour*, Eurographics (Short Paper) (2011).

[RHC08a] Damien Rohmer, Stefanie Hahmann, and Marie-Paule Cani, *Déformation interactive par skinning à volume constant*, Groupe de travail en Modélisation Géométrique (GTMG) (2008).

[RHC08b] Damien Rohmer, Stefanie Hahmann, and Marie-Paule Cani, *Local Volume Preservation for Skinned Characters*, Computer Graphics Forum (CGF). Proceedings of Pacific Graphics (PG) **27** (2008), no. 7, 1919–1927.

[RHC09] Damien Rohmer, Stefanie Hahmann, and Marie-Paule Cani, *Exact volume preserving skinning with shape control*, Eurographics/ACM SIGGRAPH Symposium on Computer Animation (SCA) (2009).

[RHC10] Damien Rohmer, Stefanie Hahmann, and Marie-Paule Cani, *Active Geometry for Game Characters*, Springer, Lecture Notes in Computer Science (LNCS). Proceedings of Motion in Games (MIG). Invited Talk. (2010).

[RPC+10] Damien Rohmer, Tiberiu Popa, Marie-Paule Cani, Stefanie Hahmann, and Alla Sheffer, *Animation Wrinkling : Augmenting Coarse Cloth Simulations with Realistic-Looking Wrinkles*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH ASIA. **29** (2010), no. 5.

[RSB96] Ari Rapporport, Alla Sheffer, and Michel Bercovier, *Volume-preserving free-form solids*, IEEE Transactions on Visualization and Computer Graphics (TVCG) **2** (1996), no. 1, 19–27.

[RSW+07] Kenneth Rose, Alla Sheffer, Jamie Wither, Marie-Paule Cani, and Boris Thibert, *Developable surfaces from arbitrary sketched boundaries*, Eurographics / ACM SIGGRAPH Symposium on Geometry Processing (SGP) (2007).

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

[SCC11] Lucian Stanculescu, Raphaëlle Chaine, and Marie-Paule Cani, *Freestyle : Sculpting Meshes with Self-Adaptive Topology*, Shape Modeling International (SMI) (2011).

[SCOL+04] Olga Sorkine, Daniel Cohen-Or, Yaron Lipman, Marc Alexa, Christian Rössl, and Hans-Peter Seidel, *Laplacian surface editing*, Eurographics / ACM SIGGRAPH Symposium on Geometry Processing (SGP) (2004).

[SF98] Karan Singh and Eugene Fiume, *Wires : a geometric deformation technique*, ACM SIGGRAPH. Computer Graphics (1998).

[SHB06] Basile Sauvage, Stefanie Hahmann, and Georges-Pierre Bonneau, *Length constrained multiresolution deformation for surface wrinkling*, Shape Modeling International (SMI) (2006), 113–121.

[SHB07] Basile Sauvage, Stefanie Hahmann, and Georges-Pierre Bonneau, *Volume preservation of multiresolution meshes*, Computer Graphics Forum (CGF). Proccedings of Eurographics **26** (2007), no. 3, 275–283.

[She] Jonathan R. Shewchuk, *Constrained Delaunay triangulations. software available at http ://www.cs.berkeley.edu/ jrs/.*

[She99] Andrei Sherstyuk, *Convolution surfaces in computer graphics*, Ph.D. thesis, School of Computer Science and Software Engineering, Monash University, Australia, 1999.

[SK04] Alla Sheffer and Vladislav Kraevoy, *Pyramid coordinates for morphing and deformation*, IEEE Symposium on 3D Data Processing, Visualization and Transmission (2004), 68–75.

[SLMB05] Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov, *ABF++ : Fast and robust angle based flattening*, ACM Transaction on Graphics (TOG) **24** (2005), no. 2.

[Sor06] Olga Sorkine, *Differential representations for mesh processing*, Computer Graphics Forum (CGF). Eurographics state of the Art **25** (2006), no. 4, 789–807.

[SP86] Thomas Sederberg and Scott Parry, *Free-form deformation of solid geometric models*, ACM SIGGRAPH **20** (1986), no. 4.

[SP04] Robert Sumner and Jovan Popovic, *Deformation transfer for triangle meshes*, ACM Transactions on Graphics (TOG), Proceedings of ACM SIGGRAPH **23** (2004), no. 3.

[Spe04] A. J. M. Spencer, *Continuum mechanics*, Dover Publications, 2004.

[SRC01] Peter-Pike J. Sloan, Charles F. Rose, and Michael F. Cohen, *Shape by example*, ACM Symposium on Interactive 3D Graphics (i3D) (2001), 135–143.

[SRS07] Eran Sharon, Benoit Roman, and Harry L. Swinney, *Geometrically driven wrinkling observed in free plastic sheets and leaves*, Physical Review **75** (2007).

[SSBT08] Thomas Stumpp, Jonas Spillmann, Markus Becker, and Matthias Teschner, *A geometric deformation model for stable cloth simulation*, Virtual Reality Interactions and Physical Simulations (VRIPHYS) (2008).

[SSGH01] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe, *Texture mapping progessive meshes*, ACM SIGGRAPH (2001).

[SZT+07] Xiaohan Shi, Kun Zhou, Yiying Tong, Mathieu Desbrun, Hujun Bao, and Baining Guo, *Mesh puppetry : Cascading optimization of mesh deformation with inverse kinematics*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[TAI+08] Kenshi Takayama, Takashi Ashihara, Takashi Ijiri, Takeo Igarashi, Ryo Haraguchi, and Kazuo Nakazawa, *A sketch-based interface for modeling myocardial fiber orientation that considers the layered structure of the ventricles*, The Journal of Physiological Sciences **58** (2008), no. 7, 487–492.

[Tau95] Gabriel Taubin, *A signal processing approach to fair surface design*, ACM SIGGRAPH, Computer Graphics (1995).

[TC09] Kai Tang and Ming Chen, *Quasi-developable mesh surface interpolation via mesh deformation*, IEEE Transactions on Visualization and Computer Graphics. **15** (2009).

[THLR98] Gabriel Taubin, William Horn, Frederic Lazarus, and Jarek Rossignac, *Geometry coding and VRML*, Proceedings of IEEE **86** (1998), no. 6, 1228–1243.

[TO99] Greg Turk and James O'Brien, *Shape transformation using variational implicit functions*, ACM SIGGRAPH, Computer Graphics (1999).

[TPPF87] Demetri Terzopoulos, John Platt, John Platt, and Kurt Fleischer, *Elastically deformable models*, ACM SIGGRAPH, Computer Graphics **21** (1987), no. 4.

[TPS09] Bernhard Thomaszewski, Simon Pabst, and Wolfgang Strasser, *Continuum-based strain limiting*, Computer Graphics Forum (CGF). Proceedings of Eurographics **28** (2009), no. 2, 569–576.

[TWB+07] Emmanuel Turquin, Jamie Wither, Laurence Boissieux, Marie-Paule Cani, and John F. Hughes, *A Sketch-Based Interface for Clothing Virtual Characters*, IEEE Computer Graphics and Applications (CG&A) **27** (2007), no. 1, 72–81.

[vFTS06] Wolfram von Funck, Holger Theisel, and Hans-Peter Seidel, *Vector field based shape deformations*, ACM Transactions on Graphics (TOG). Proceedings of ACM SIGGRAPH **25** (2006), no. 3.

[vFTS07] Wolfram von Funck, Holger Theisel, and Hans-Peter Seidel, *Elastic secondary deformations by vector field integration*, Eurographics / ACM SIGGRAPH Symposium on Geometry Processing (SGP) (2007), no. 3.

[VT98] Pascal Volino and Nadia Magnenat Thalmann, *Virtual Clothing. Theory and Practice*, Springer, 1998.

[Wan08] Charlie C. Wang, *Toward Flattenable Mesh Surfaces*, Computer Aided Design (CAD) **40** (2008).

[WDAH10] Tim Winkler, Jens Drieseberg, Marc Alexa, and Kai Hormann, *Multi-scale geometry interpolation*, Computer Graphics Forum (CGF). Proceedings of Eurographics **29** (2010), no. 2, 309–318.

[WHRO10] Huamin Wang, Florian Hecht, Ravi Ramamoorthi, and James F. O'Brien, *Example-based wrinkle synthesis for clothing animation*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **29** (2010), no. 4.

[WL08] Rich Wareham and Joan Lasenby, *Bone glow : An improved method for the assignment of weights for mesh deformation*, Conference on Articulated Motion and Deformable Objects (AMDO) (2008).

[WMKG07] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun, *Discrete laplace operators : No free lunch*, Eurographics / ACM SIGGRAPH Symposium on Geometry Processing (SGP) (2007).

[WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill, *Data structure for soft objects*, The Visual Computer **2** (1986), no. 4, 227–234.

[WOR10] Huamin Wang, James O'Brien, and Ravi Ramamoorthi, *Multi-resolution isotropic strain limiting*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH Asia **29** (2010), no. 6.

[WP02] Xiaohuan C. Wang and Cary Philips, *Multi-weight enveloping : Least-squares approximation techniques for skin animation*, ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA) (2002).

[WSLG07] Ofir Weber, Olga Sorkine, Yaron Lipman, and Graig Gotsman, *Context-Aware Skeletal Shape Deformation*, Computer Graphics Forum (CGF), Proceedings of Eurographics **26** (2007), no. 3.

[WT04] Charlie Wang and Kai Tang, *Achieving developability of a polygonal surface by minimum deformation : A study of global and local optimization approaches*, The Visual Computer. **20** (2004).

[WvO96] Bryan Wyvill and Kees van Overveld, *Tiling techniques for implicit skeletal models*, ACM SIGGRAPH Course Notes (1996).

[XZB06] Dong Xu, Hongxin Zhang, and Hujun Bao, *Non-uniform differential mesh deformation*, Lecture Notes in Computer Science (LNCS). Proceedings of Computer Graphics International **4035** (2006), 54–65.

[XZWB05] Dong Xu, Hongxin Zhang, Qing Wang, and Hujun Bao, *Poisson Shape Interpolation*, ACM Symposium on Solid and Physical Modeling (2005), 267–274.

[YHB05] Alex Yvart, Stefanie Hahmann, and Georges-Pierre Bonneau, *Hierarchical triangular splines*, ACM Transactions on Graphics **24** (2005), no. 4, 1374–1391.

[YPP07] Robert Y. Yang, Kari Pulli, and Jovan Popovic, *Real-time enveloping with rotational regression*, ACM Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **26** (2007), no. 3.

[YZ05] X. S. Yang and Jian Zhang, *Realistic skeleton driven skin deformation*, International Conference on Computational Science and its Applications (ICCSA) (2005), 1109–1118.

[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 Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **23** (2004), no. 3.

[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 Transaction on Graphics (TOG). Proceedings of ACM SIGGRAPH **24** (2005), no. 5.

[ZSD+00] Denis Zorin, Peter Schröder, Tony DeRose, Leif Kobbelt, Adi Levin, and Wim Sweldens, *Subdivision for modeling and animation*, ACM SIGGRAPH Course Notes (2000).

[ZSS97] Denis Zorin, Peter Schröder, and Wim Sweldens, *Interactive multiresolution mesh editing*, ACM SIGGRAPH, Computer Graphics (1997).

[ZXTD10] Kun Zhou, Weiwei Xu, Yiying Tong, and Mathieu Desbrun, *Deformation transfer to multi-component objects*, Computer Graphics Forum (CGF), Proceedings of Eurographics **29** (2010), no. 2, 319–325.

[ZYP06] Wen Zheng, Jun-Hai Yong, and Jean-Claude Paul, *Simulation of Bubbles*, Eurographics/ACM SIGGRAPH Symposium on Computer Animation (SCA) (2006).