HAL
open science

Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages

Caroline Larboulette

To cite this version:

Caroline Larboulette. Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages. Modélisation et simulation. INSA de Rennes, 2004. Français. NNT : . tel-00528651

HAL Id: tel-00528651

https://theses.hal.science/tel-00528651v1

Submitted on 22 Oct 2010

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

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

N° d'ordre: D-04-13

THÈSE

présentée devant

l'Institut National des Sciences Appliquées de Rennes

pour obtenir

le titre de :
DOCTEUR DE L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES
Mention INFORMATIQUE

par

Caroline LARBOULETTE

Équipes d'accueil : SIAMES - EVASION
École Doctorale : Matisse

Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages

Encadrée par

Mr	Bruno	ARNALDI	Directeur
Mme	Marie-Paule	CANI	Co-directrice

Soutenue le 30 Novembre 2004 devant la commission d'examen

Mr	Jean-Louis	PAZAT	Président
Mr	Samir	AKKOUCHE	Rapporteurs
Mr	Yannick	RÉMION	
Mr	Marc	NEVEU	Examinateurs
Mr	Jean	CAMILLERAPP	
Mme	Marie-Paule	CANI	

N° d'ordre: D-04-13

THÈSE

présentée devant

l'Institut National des Sciences Appliquées de Rennes

pour obtenir

le titre de :
DOCTEUR DE L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES
Mention INFORMATIQUE

par

Caroline LARBOULETTE

Équipes d'accueil : SIAMES - EVASION
École Doctorale : Matisse

Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages

Encadrée par

Mr	Bruno	ARNALDI	Directeur
Mme	Marie-Paule	CANI	Co-directrice

Soutenue le 30 Novembre 2004 devant la commission d'examen

Mr	Jean-Louis	PAZAT	Président
Mr	Samir	AKKOUCHE	Rapporteurs
Mr	Yannick	RÉMION	
Mr	Marc	NEVEU	Examinateurs
Mr	Jean	CAMILLERAPP	
Mme	Marie-Paule	CANI	

Remerciements

Je souhaite tout d'abord remercier mes directeurs de thèse, Bruno et Marie-Paule pour m'avoir permis d'effectuer cette thèse au sein de leurs équipes de recherche respectives, *SIAMES* et *EVASION*, et d'avoir ainsi pu profiter de deux façons de travailler différentes. Que ce soit l'une ou l'autre des deux équipes, l'accueil y a été chaleureux et les séjours très enrichissants, tant sur le plan de la recherche que sur le plan humain.

Je remercie également Samir Akkouche et Yannick Rémion pour avoir accepté d'être les rapporteurs de ma thèse, et Jean-Louis Pazat pour m'avoir fait l'honneur de présider mon jury.

Je tiens aussi à remercier Marc Neveu et Jean Camillerapp pour avoir accepté de faire partie de mon jury, malgré les changements tardifs de composition du jury.

Je tiens également à remercier Christine Depraz pour m'avoir apporté un point de vue artistique sur mon travail et aussi pour avoir créé une grande partie des animations présentées dans cette thèse.

Je ne citerais pas les noms des nombreux thésards, ingénieurs, chercheurs ou stagiaires des deux équipes que j'ai rencontrés pendant ces trois ans, mais je les remercie pour tous les bons moments passés avec eux. Ces remerciements s'adressent bien évidemment aussi à tous les membres de l'équipe *ARTIS*.

Enfin, je remercie ma famille et mes proches pour leur soutien et leur enthousiasme qui a été constamment présent durant ces trois années.

SOMMAIRE

1 Introduction **5**

I État de l'art **9**

2 Modélisation d'objets déformables structurés et animation traditionnelle par squelette **13**

3 Modélisation et déformation des muscles, tissus et peau de personnages **39**

II Contributions **77**

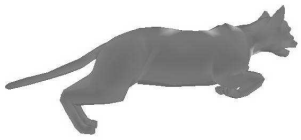
4 Skinning dynamique **81**

5 Plis dynamiques en temps-réel **107**

6 Combinaison des deux outils **129**

7 Discussion sur l'intérêt et les limitations des méthodes géométriques **135**

8 Conclusion et travaux futurs **139**



CHAPITRE 1

Introduction

Et alors la marmotte, elle met le chocolat dans le papier d'alu . . .

Le randonneur, publicité Milka

Les personnages virtuels sont de plus en plus présents dans notre vie quotidienne. Après avoir envahi le domaine du jeu vidéo, on les retrouve maintenant au cinéma[1], sur notre petit écran[2], et bientôt ils peupleront des mondes virtuels[3] où seront immergés des utilisateurs réels. Cependant, ces activités nécessitent une immersion parfaite de l'utilisateur qui ne peut être obtenue que grâce à des personnages de synthèse réalistes. La difficulté est variable selon le type de personnage que l'on cherche à représenter. En effet, s'il est facile de rendre un personnage imaginaire crédible aux yeux de l'utilisateur, la tâche est plus délicate en ce qui concerne un humanoïde, l'œil humain[4] étant entraîné à voir de réels humains et percevant directement les moindres défauts. Parallèlement à l'animation d'humanoïdes, on retrouve beaucoup de personnages, des animaux la plupart du temps, auxquels on attribue des actions typiquement humaines. C'est le cas des personnages de publicités tels que la fameuse marmotte de Milka ou de nombreux personnages de dessins animés ou de films d'animation. Bien que n'ayant de réalité que dans l'imaginaire, ces personnages doivent être *crédibles*, c'est à dire que leurs déformations doivent être naturelles et comporter les mêmes effets que les animations réalistes.

[1] Comme dans Shrek, Final Fantasy ou plus récemment Immortel.

[2] Dans les publicités essentiellement, telles que celles de la Caisse d'Épargne (chiens et éléphants) ou du chocolat Milka (les marmottes).

[3] Voir le musée virtuel de l'équipe SIAMES de Rennes ou le projet *Virtual life in Pompeii* du MIRALab, Genève.

[4] En réalité, c'est le cerveau qui analyse les données transmises par l'œil.

Motivations

Cependant, le réalisme souhaité au sein des animations, aussi bien au niveau des mouvements qu'au niveau des déformations de la surface externe, entre directement en conflit avec les temps de calcul nécessaires pour les produire. Intuitivement, on comprend bien que plus les temps de calcul sont élevés, plus le résultat est beau, conforme à ce que l'on souhaitait obtenir, de la même façon qu'un artiste passe plus de temps à réaliser un portrait à l'huile qu'une prise de croquis. Malheureusement, si l'on considère la production de films, même si l'on peut passer des jours à calculer quelques secondes d'animation, il est souhaitable que cette durée ne soit pas trop excessive. De plus, l'utilisation d'outils temps-réel peut permettre d'obtenir rapidement une idée du résultat avant de lancer de lourds calculs. Ceci est d'autant plus vrai dans le domaine du jeu vidéo où les animations doivent nécessairement être produites en temps-réel. Actuellement, la réduction des temps de calcul se fait par une simplification des modèles, en commençant par la suppression des détails comme les plis ou les effets dynamiques. Cette simplification est tellement extrême dans les jeux vidéos que les personnages finissent par ne plus être réalistes.

Contexte

Dans le cadre de cette thèse, nous souhaitons apporter des solutions à l'animation temps-réel de personnages et en particulier en améliorer le réalisme par l'ajout de détails. Notre application principale étant le jeu vidéo, nous montrerons dans l'état de l'art pourquoi notre choix s'est très vite porté sur l'animation classique de personnages modélisés par maillage polygonal et pourvus d'un squelette interne rigide. Le squelette articulé est animé d'un mouvement rigide créé à la main par un animateur ou bien en utilisant de la cinématique inverse, de la capture du mouvement ou bien encore résultant d'un scénario d'analyse comportementale. Les déformations statiques des couches muscles, tissus et peau sont animées de manière traditionnelle, par *skinning*[5].

Objectifs

Notre objectif est d'ajouter à une animation existante, des effets qui permettent d'en augmenter le réalisme ou bien au contraire, d'exagérer certaines déformations dans le cas d'un personnage de dessin animé par exemple. Les outils que nous souhaitons mettre en place doivent donc pouvoir s'ajouter comme des couches supplémentaires du modèle, sans demander à l'utilisateur de modifier l'existant et sans alourdir les temps de calcul. Nous avons dégagé deux principaux manques au niveau de ces animations : les effets d'inertie des masses musculaires et graisseuses lors de mouvements secs et rapides ainsi que les plis dynamiques de la peau ou des vêtements au niveau des articulations.

Contributions

Notre première contribution s'attache à ajouter, en temps-réel, des déformations dynamiques de la couche muscles/graisse aux déformations statiques obtenues par *skinning* ou tout autre technique équivalente. L'effet dynamique est calculé à travers une simplification des lois de l'élasticité linéaire. Pour cela, des éléments dynamiques associant les vibrations d'un ressort à un ensemble de poids

[5] Le *skinning* est une technique traditionnelle d'habillage d'un squelette qui sera présentée en détails à la section 2.3.1.2.

assignés aux sommets du maillage permettent d'effectuer un *skinning dynamique*. L'emplacement, la fréquence et l'amplitude de ces effets sont spécifiés, de manière très simple, par l'utilisateur.

Sur cette couche vient se greffer une surface fine qui peut être soit la peau, soit directement un vêtement. Cette surface, d'aire quasiment constante, suit les déformations des muscles et des tissus graisseux tout en se plissant lorsqu'elle se retrouve compressée. Pour cela, l'utilisateur positionne et oriente une courbe de contrôle qui se déforme à longueur constante et spécifie sa région d'influence. Pendant l'animation, des plis géométriques sont générés en temps-réel dans les zones couvertes par l'outil, le maillage étant raffiné localement à la volée.

Organisation du document

La suite de ce document se décompose en deux parties. La première dresse un état de l'art de l'animation de personnages alors que la seconde présente nos contributions et nos résultats.

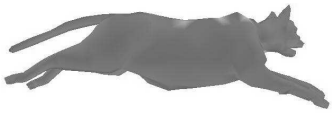
Le chapitre 2 s'attache à décrire les modèles géométriques, les modèles d'animation et les techniques de déformation d'objets déformables structurés. Ce chapitre sera le plus général possible sur ces techniques et permettra de mieux comprendre les algorithmes utilisés dans les travaux antérieurs pour modéliser et animer les muscles, les tissus adipeux et la peau de personnages, présentés au chapitre 3. Un bilan de l'existant conclura cette partie en montrant en quoi les travaux antérieurs ne permettent pas d'atteindre les buts que nous nous sommes fixés.

Le chapitre 4 propose une technique permettant d'ajouter des effets dynamiques aux muscles et aux tissus adipeux d'un personnage. Basé sur le *skinning* classique, le *skinning dynamique* utilise des éléments de physique simplifiés et ne nécessite qu'un très rapide pré-calcul. Le chapitre 5 présente une méthode permettant de rajouter des plis de la peau et/ou de vêtements à un personnage animé. Ces contributions sont temps-réel et utilisables conjointement ou indépendamment sur une animation existante, comme le montre le chapitre 6. Le chapitre 7 discute les avantages et les limitations des méthodes géométriques que nous avons développées par rapport aux méthodes anatomiques ou physiques existantes tout en montrant le besoin de telles méthodes en ce qui concerne les applications temps-réel comme les jeux vidéos.

Enfin, le chapitre 8 conclut, tout en s'élargissant sur les travaux futurs envisagés à court terme et à long terme.

Première partie

État de l’art



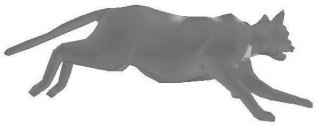
Introduction

La première partie de cette thèse propose un état de l'art en deux chapitres. Le premier, très général, s'attache à décrire les procédés classiques de mise en place d'une animation d'objet déformable structuré. Pour cela, un tour d'horizon des représentations surfaciques existantes, de l'utilisation d'un squelette en animation et des techniques de déformations géométriques et physiques permettra d'éclairer le lecteur novice en animation.

Le second chapitre se veut plus spécifique sur l'animation de personnages. À travers un large parcours des articles parus ces trente dernières années, il retrace l'évolution des techniques d'animation de personnages en décrivant les solutions existantes pour résoudre les problèmes classiques rencontrés à différents niveaux. En effet, nos efforts se sont concentrés sur les techniques de déformations statiques et dynamiques des parties non rigides du corps d'un personnage telles que les muscles, les tissus adipeux ou bien la peau.

Par déformation *statique*, nous entendons toutes les déformations créées par des modèles descriptifs, c'est à dire lorsque la déformation est une forme modélisant un état d'équilibre des parties déformables d'une pose donnée. Dans ce cas, l'animateur peut *décrire* le résultat qu'il veut obtenir, et c'est le type de déformations que l'on obtiendra par *skinning* après avoir spécifié une série de positions clefs par exemple.

Les déformations dynamiques quant à elles proviennent de modèles générateurs de déformations, qui, en décrivant les *causes* d'un phénomène, permettent de simuler tous les états intermédiaires de la déformation avant d'aboutir éventuellement à une forme statique d'équilibre. Typiquement, ce sont tous les phénomènes vibratoires obtenus lors de mouvements rapides et secs, comme par exemple, le ballottement des cuisses ou du ventre d'un personnage qui court. Ces déformations ne peuvent pas être décrites par des modèles purement géométriques et font donc appel à des modèles physiques plus ou moins complexes tels que les systèmes masses-ressorts ou les éléments finis.



CHAPITRE 2

Modélisation d'objets déformables structurés et animation traditionnelle par squelette

Tout au long de cette thèse, nous employons le terme d'*objet structuré* pour désigner tout objet $3D$ constitué d'une partie rigide animée d'un mouvement et d'une partie non rigide qui se déforme statiquement et/ou dynamiquement en réponse au mouvement de la structure rigide. Un tel objet sera, par exemple, la représentation d'un personnage humain, la structure rigide étant le squelette et les parties déformables étant les muscles, les tissus adipeux et la peau.

La réalisation d'une animation complexe faisant évoluer plusieurs de ces objets déformables structurés se fait en plusieurs étapes. Il faut tout d'abord créer un modèle géométrique pour chaque objet (c'est à dire une enveloppe), lui associer un squelette d'animation (étape de *rigging*) puis animer le squelette et déformer l'objet en fonction du mouvement.

La première partie du chapitre présente les différents modèles couramment utilisés dans les domaines de la recherche et de l'industrie pour représenter l'enveloppe d'un objet $3D$. La seconde section présente les solutions existantes permettant d'animer un squelette rigide de personnage. Enfin, la troisième section fait le lien entre les deux premières en décrivant différentes techniques géométriques et physiques de déformation d'un modèle en réponse à la déformation du squelette interne.

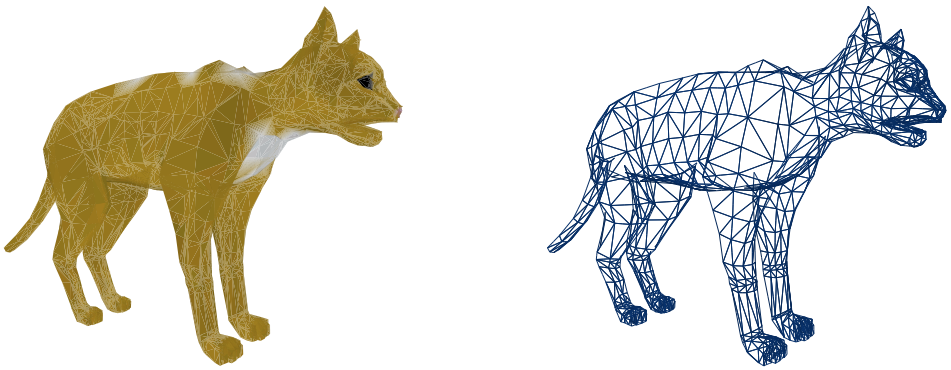
2.1 Représentation de l'enveloppe d'un objet 3D

Le modèle géométrique est une représentation mathématique d'un objet $3D$ sur lequel sont appliquées des déformations lors de l'animation, soit directement, soit à travers une représentation

intermédiaire. Dans la suite, nous présentons quatre types de surfaces couramment utilisées pour représenter l'enveloppe géométrique d'un objet $3D$: les surfaces polygonales, les surfaces paramétriques, les surfaces de subdivision et les surfaces implicites. Chaque type de surface possède ses caractéristiques propres et le choix de l'une ou l'autre de ces surfaces dépend des contraintes qui doivent être satisfaites par l'application. Le choix du modèle géométrique joue notamment un rôle essentiel lors de la détection de collisions entre plusieurs objets ou entre différentes parties d'un même objet, ainsi que lors de l'affichage à l'écran.

2.1.1 Surfaces polygonales

Les surfaces polygonales sont constituées d'un ensemble de primitives géométriques planes, le plus souvent des triangles ou des quadrilatères, décrivant une surface $3D$ (un volume dans le cas d'une surface fermée). Ce ne sont rien d'autre qu'une collection de plans délimités par des contours, et, de ce fait, leur expression mathématique est très simple[1]. Les polygones constituant l'objet étant connectés entre eux, on parle le plus souvent de *maillage polygonal*. Le maillage triangulaire est la forme la plus simple de représentation d'un objet, en ce sens qu'un tel maillage est directement affichable par une carte graphique. Cette rapidité d'affichage rend ce type de surfaces très utilisé dans les jeux vidéos. Un exemple de modèle ainsi composé d'un ensemble de facettes est montré sur la figure 2.1. Le chat est formé d'un ensemble de triangles qui approchent au mieux la forme de l'objet $3D$ (b) et qui sont colorés lors de l'affichage de façon à ce que la surface apparaisse lisse (a).



(a) (b)

FIG. 2.1 – Un modèle de chat : (a)-le modèle affiché à l'écran ; (b)-affichage *fil de fer* qui montre les triangles le composant.

Cependant, si cette représentation est rapide à afficher, elle n'est pas la plus simple à utiliser pour l'infographiste lors des étapes de modélisation ou d'animation. En effet, lorsqu'il faut déformer le modèle créé, l'animateur doit manipuler les points du maillage. Cependant, nous verrons à la section 2.3 que certaines techniques existent pour alléger le travail de l'animateur.

2.1.2 Surfaces paramétriques

Les surfaces paramétriques sont très utilisées en modélisation car elles offrent un outil de dessin intuitif et produisent des surfaces plus lisses que les maillages polygonaux, pour un même nombre de points de contrôle. Si l'on se place dans le cas $1D$, pour dessiner une courbe lisse, l'infographiste ne

[1] Équation du plan : $ax + by + cz + d = 0$

spécifie que quelques points et une courbe approchant ou interpolant ces points est automatiquement calculée. Ces courbes peuvent être de différents types, suivant la fonction mathématique utilisée pour l'approximation ou l'interpolation. Les splines sont les plus couramment utilisées bien qu'il en existe d'autres, comme par exemple les courbes de Bézier.

Nous allons tout d'abord nous intéresser de plus près au fonctionnement de ces courbes splines (cas $1D$) en montrant les différents types de splines et leurs équations, avant d'étendre ces résultats aux surfaces paramétriques, cas à 2 paramètres.

Courbes splines

Une spline est une courbe définie par des points de contrôle. Par exemple, dans le cas d'une B-spline cubique, c'est une fonction polynomiale par morceaux à laquelle on assure certaines conditions de continuité au niveau des recollements des portions de courbe. Chaque segment de courbe polynomiale est défini par un petit nombre de points de contrôle voisins, ce qui fait que la courbe est toujours définie localement. Ainsi, lorsqu'un point de contrôle est modifié, la courbe est modifiée *localement* avec des temps de calcul peu coûteux.

Il existe différents types de courbes splines selon leurs propriétés. On distingue tout d'abord les splines d'interpolation des splines d'approximation. En effet, dans le premier cas, la courbe passe par les points de contrôle P_i et est constituée de segments cubiques se recollant de manière C^1 en chaque point. Un exemple de ces splines sont les splines cardinales présentées à la figure 2.2 (a) et (c).

Dans le cas des splines d'approximation, la courbe ne passe pas par les points de contrôle mais est incluse dans le polygone englobant formé par ces points. Un avantage de cette propriété peut être par exemple, une détection de collisions facilitée. Les splines d'approximation les plus simples sont les B-splines cubiques uniformes présentées à la figure 2.2 (b) et (d). Constituée d'une combinaison linéaire de fonctions de base B_i de degré au moins trois[2], une B-spline $C(u)$ aura pour équation :

$$C(u) = \sum_{i=0}^{n} B_i(u) P_i$$

avec :
P_i, les points de contrôle,
B_i, les fonctions de bases,
$\sum_{i=0}^{n} B_i(u) = 1$ pour tout u.

Enfin, il est possible de pondérer l'influence de chaque point de contrôle. Ainsi, la B-spline rationnelle non uniforme (NURBS en anglais) est définie par des points de contrôle, mais aussi par des poids w_i associés à chaque point P_i. Cela permet à chaque point de contrôle d'*attirer* plus ou moins la courbe à lui. Dans ce cas, les segments de courbe ne sont plus définis par des polynômes mais par des fractions rationnelles de polynômes. L'équation d'une courbe NURBS $C(u)$ prend alors la forme suivante :

$$C(u) = \frac{\sum_{i=0}^{n} w_i B_i(u) P_i}{\sum_{i=0}^{n} w_i B_i(u)}$$

avec :
P_i les points de contrôle,
B_i les fonctions de base normalisées de degré k,
w_i les poids.

[2]En pratique, les B-splines cubiques (degré 3) sont les plus utilisées car elle apportent une continuité C^2 entre les portions de courbe pour des temps de calcul raisonnables.

Un rapide coup d'œil à ces équations nous montre l'évidente complexité mathématique des courbes paramétriques comparées aux surfaces polygonales. De cette complexité découlent des temps de calcul légèrement plus longs pour l'affichage. Bien évidemment, ces équations sont transparentes à l'infographiste, qui, lors de la modélisation, se contente de placer ses points de contrôle. Ensuite, pour ajuster la forme de la courbe, il peut déplacer les points de contrôle, en ajouter ou en enlever ou bien changer leur poids.

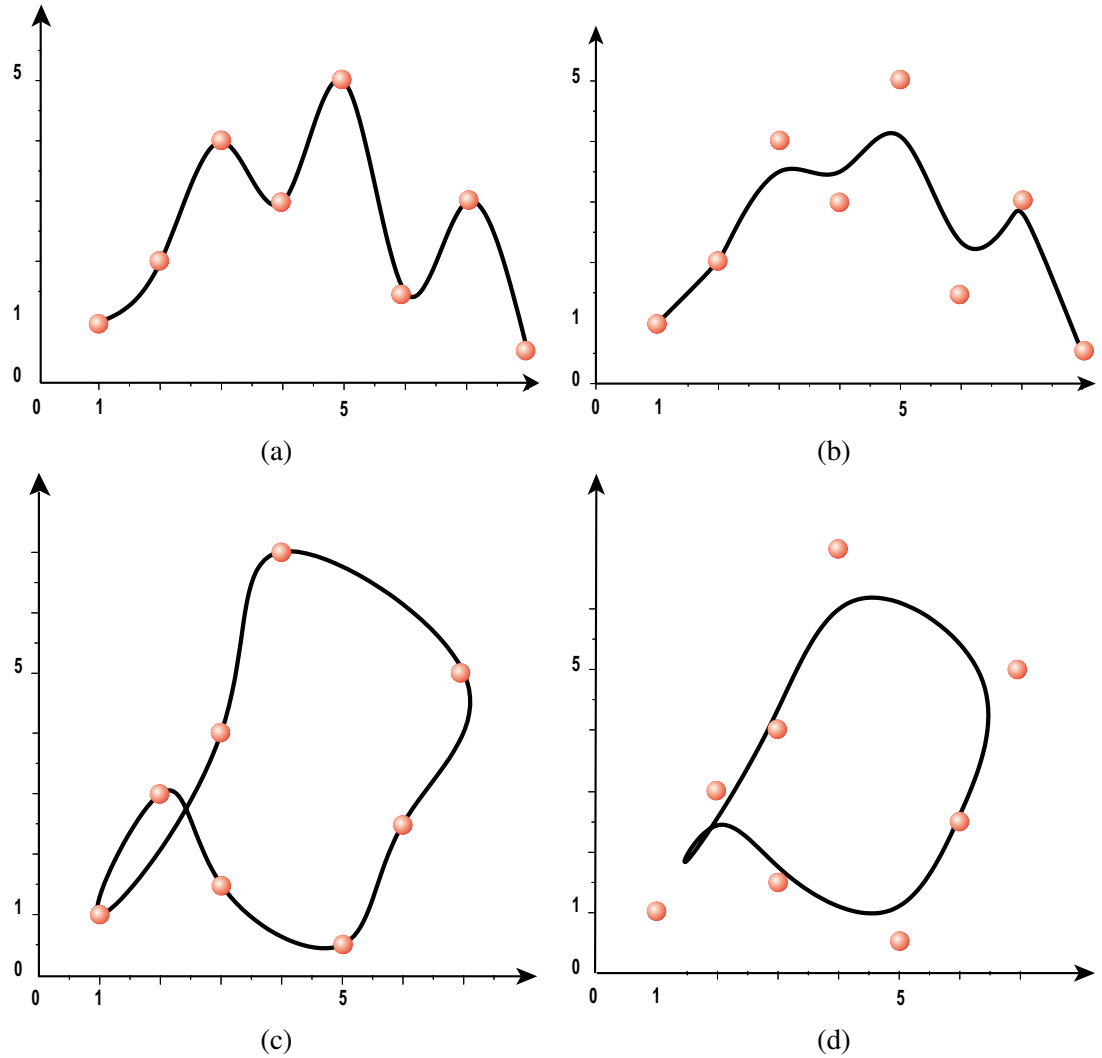
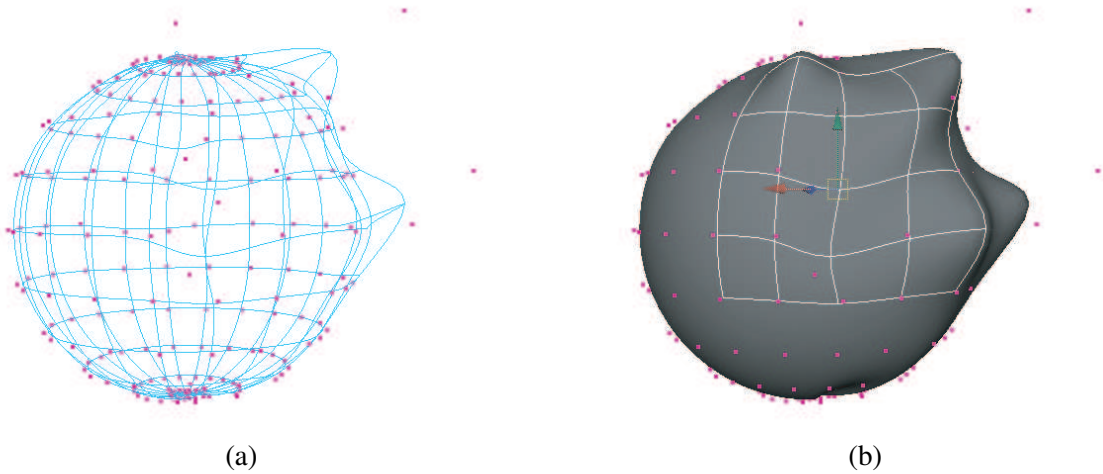


FIG. 2.2 – Différents exemples de courbes splines construites à partir des mêmes points de contrôle : (a)-spline d'interpolation ; (b)-B-spline ; (c)-spline cubique fermée ; (d)-B-spline cubique fermée.

Carreaux de surface

On peut étendre les courbes splines à des surfaces splines bi-paramétriques qui permettent de créer des surfaces $3D$. Il s'agit d'une extension des courbes par augmentation de la dimension de l'espace paramétrique. Le *patch*, ou carreau de surface, est alors l'équivalent du segment de courbe d'une spline simple. Le fonctionnement des surfaces splines reste le même que pour les courbes, sauf que le domaine paramétrique est de dimension 2 et le résultat est une surface et non plus une courbe (voir figure 2.3).



(a) (b)

FIG. 2.3 – Un exemple de surface $3D$ créée à partir de NURBS : (a)-les points de contrôle et les courbes NURBS définies par ces points ; (b)-la surface finale et l'influence locale d'un point de contrôle.

Les surfaces paramétriques sont devenues le standard chez les infographistes pour représenter des surfaces lisses de géométrie complexe. En particulier, les logiciels commerciaux tels que Maya [ali] ou 3ds Max [3ds] utilisent des surfaces NURBS (Non-Uniform Rational B-Spline). En effet, les avantages de ces surfaces sont nombreux :

- la fonction mathématique définissant l'enveloppe est de la même forme, que ce soit pour des formes analytiques standard (sphère, cylindre) ou des formes libres ;
- le rendu se fait suffisamment rapidement, par des algorithmes stables et efficaces de conversion en triangles ;
- les NURBS sont invariantes aux transformations affines et perspectives ;
- elles sont une généralisation des B-splines non rationnelles et des surfaces de Bézier.

Pour déformer les modèles paramétriques, il faut déplacer les points de contrôle. L'avantage est que l'action d'un point de contrôle est locale (voir figure 2.3 (b)) et que l'information visuelle est disponible en temps-réel. En effet, les calculs étant peu coûteux, lorsqu'un point est modifié, le résultat de la modification est immédiatement visible à l'écran et l'utilisateur peut donc comparer le résultat produit au résultat attendu. Cependant, dans le cas de la modélisation d'objets complexes (imaginons un corps humain), l'infographiste peut être très vite découragé par le nombre de points nécessaires à la représentation de l'objet et donc leur manipulation. Un utilisateur non expérimenté sera vite déçu s'il n'est pas très habile.

À cet inconvénient s'ajoutent des défauts intrinsèques aux surfaces paramétriques comme le manque de souplesse au niveau de l'ajout de détails ou l'impossibilité de créer des surfaces de topologie originale[3]. Le raccordement entre deux morceaux de surfaces paramétriques est possible le long d'un bord mais la création d'embranchements pose des problèmes, dans le cas de la jonction entre deux morceaux de surfaces qui n'ont pas le même nombre de carreaux par exemple. Une solution est proposée dans [Can90]. Une autre source de problèmes est l'ajout de détails. En effet, pour déformer localement la surface, il est souvent nécessaire de rajouter des points de contrôle. Malheureusement, on ne peut pas rajouter un point de contrôle indépendamment du reste de la surface, ce qui se traduit par le fait qu'il faut rajouter une ligne et une colonne complètes de points de contrôle

[3]Les topologies représentables sont limitées aux plans, cylindres ou tores.

alors qu'un seul aurait été nécessaire (voir figure 2.4). Ce problème peut cependant être traité par l'utilisation de splines hiérarchiques introduites par Forsey en 1988 [FB88]. Ces techniques ne sont malheureusement pas encore accessibles dans les logiciels du commerce.

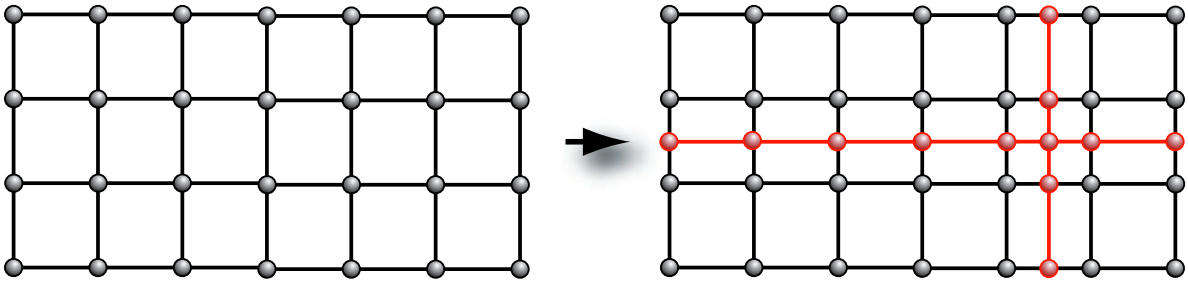


FIG. 2.4 – L'ajout d'un point de contrôle sur une surface paramétrique entraîne automatiquement l'ajout d'une colonne et d'une ligne entières de points de contrôle car il faut découper chaque domaine paramétrique.

L'utilisation de surfaces paramétriques peut aussi poser des problèmes lors de leur animation, notamment au niveau de la détection de collisions. En effet, l'équation d'une courbe est un polynôme de degré au moins trois et il n'est pas facile de résoudre mathématiquement l'intersection ou non de deux courbes, *a fortiori*, de deux surfaces.

2.1.3 Surfaces de subdivision

Les surfaces de subdivision sont définies comme la limite d'un processus infini de raffinement. C'est un processus récursif qui agit sur un maillage polygonal en subdivisant les polygones à chaque étape et en calculant des positions pour les nouveaux sommets créés et pour les anciens de façon à ce que la surface tende vers une surface lisse. Le gros avantage des surfaces de subdivision sur les surfaces paramétriques est que la surface créée peut être de topologie quelconque et les embranchements ne posent aucun problème.

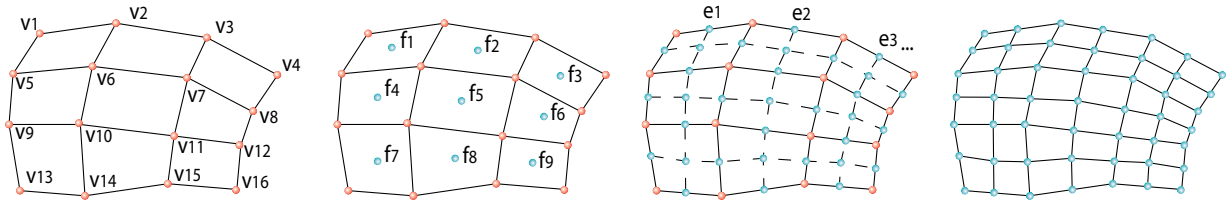
Afin d'illustrer le principe de la subdivision récursive, nous allons appliquer un schéma bien connu sur un maillage polygonal : le schéma de Catmull-Clark [CC78].

Soit un maillage initial, composé de quadrilatères grossiers dont les sommets sont notés v_i (voir figure 2.6 (a)). Une étape de subdivision s'effectue en ajoutant tout d'abord un nouveau sommet f_i au milieu de chaque face. La position de ce nouveau sommet s'exprime entièrement à partir des positions des 4 sommets v_i définissant la face. Ainsi, si l'on se réfère à la figure 2.6 (b), la position du sommet f_1 sera égale à $\frac{v_1+v_2+v_5+v_6}{4}$.

Ensuite, chaque arête est coupée en deux. Le nouveau sommet e_i est égal à la moyenne du centre de l'arête et de la moyenne des deux sommets f_i des deux faces partageant cette arête (figure (c)). Chaque nouveau point f_i est alors connecté aux nouveaux points e_i de son ancienne face.

FIG. 2.5 – Geri, personnage principal du film d'animation *Geri's Game*, a été modélisé par des surfaces de subdivision de type Catmull-Clark.

Enfin, les anciens sommets v_i sont déplacés à la position $\frac{Q+2R+S}{4}$, Q étant la moyenne des nouveaux points f_i adjacents à v_i, R la moyenne des milieux des anciennes arêtes incidentes à v_i, et S, l'ancienne position du sommet v_i (voir figure (d)).

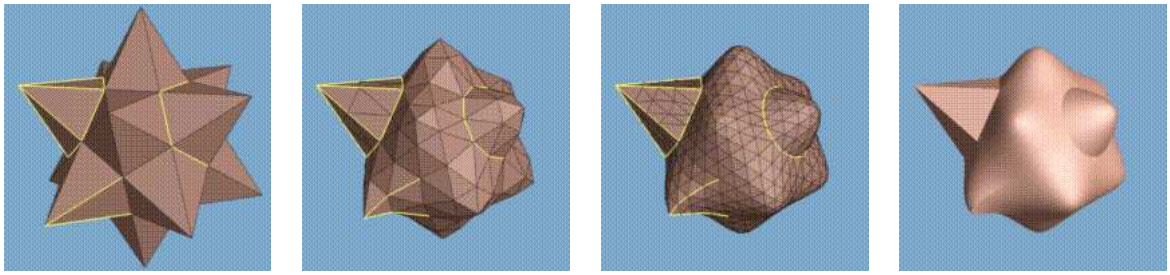


(a) (b) (c) (d)

FIG. 2.6 – Un exemple didactique de subdivision par le schéma de Catmull-Clark : (a)-le maillage original composé de sommets v_i ; (b)-ajout de nouveaux sommets f_i au centre de chaque face ; (c)-ajout de nouveaux sommets e_i au centre de chaque arête et de nouveaux liens entre les nouveaux sommets f_i et e_i ; (d)-déplacement des anciens sommets pour obtenir le maillage final.

Le maillage obtenu en (d) est donc plus fin et plus lisse que le maillage original en (a), mais il garde la même topologie et le même type de polygones (des quadrilatères). Il est donc possible d'appliquer immédiatement une nouvelle étape de subdivision pour raffiner encore le maillage.

Par exemple, si l'on se réfère à la figure 2.7, en (a), on peut voir le maillage de contrôle (c'est à dire, ce que l'infographiste dessine réellement), en (b), le maillage obtenu après un raffinement en utilisant le schéma de Loop [Loo87], en (c), celui obtenu après deux raffinements et enfin en (d) après un nombre suffisant de raffinements.



(a) (b) (c) (d)

FIG. 2.7 – Une surface de subdivision et son processus de raffinement : (a)-maillage initial ; (b)-après une étape de subdivision ; (c)-après deux étapes ; (d)-après n étapes (images extraites de [HDD +94]).

Il existe différents algorithmes de raffinement qui opèrent sur différents types de maillages (triangulaire ou quadrangulaire), en divisant soit les faces, soit les sommets (voir figure 2.8).

Dans le cas de la subdivision de faces, le processus est qualifié de di-adique car chaque arête est coupée en deux à chaque étape de la subdivision. Étant donné que la position du point résultant dépend de ses voisins, il n'est pas possible de résoudre plusieurs étapes de subdivision d'un seul coup. D'autre part, cela oblige à stocker la connectivité des polygones pendant la phase de modélisation. Dans le cas où l'on applique un algorithme de subdivision sur une soupe de polygones, le processus est long en temps de calcul, vu que la connectivité des polygones est à recalculer.

Le tableau 2.1 résume les algorithmes de subdivision les plus courants en leur associant le masque

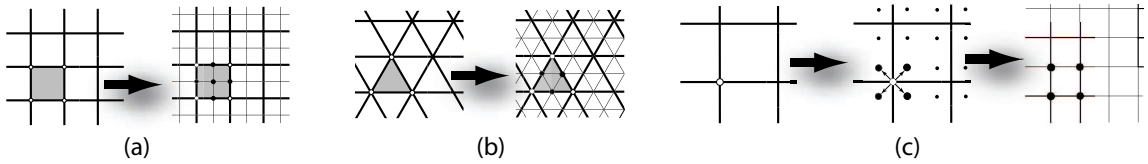
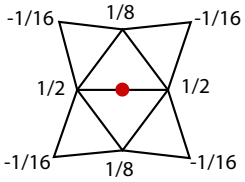
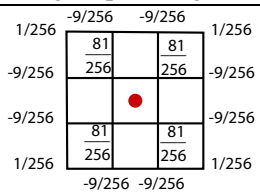
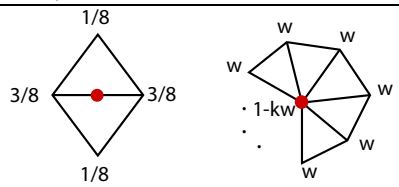
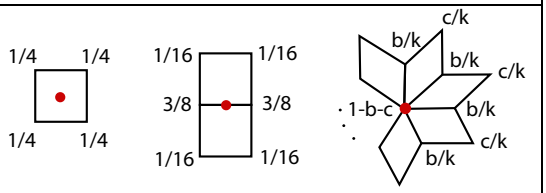


FIG. 2.8 – Différents principes de subdivision existent : (a)-division de quadrilatères ; (b)-division de triangles ; (c)-division des sommets (images extraites de [ZSD +03]).

	Maillages triangulaires	Maillages quadrangulaires
Interpolation	 Butterfly modifié [DLG90, ZSS96] (C^1)	 Kobbelt [Kob96](C^1)
Approximation	 Loop [Loo87] (C^2)	 Catmull-Clark [CC78] (C^2)

TAB. 2.1 – Tableau récapitulatif des schémas de subdivision les plus courants, interpolants ou approximants, applicables à des triangles ou des quadrilatères (données extraites de [ZSD +03]).

de subdivision à utiliser dans le cas général. En effet, il existe des cas particuliers qui sont traités différemment, par exemple, lorsqu'un sommet n'a pas le nombre requis de voisins ou bien se trouve sur un bord du maillage. Cela permet de calculer des nouveaux points, quelle que soit la topologie du maillage et le nombre de carreaux voisins.

L'avantage des surfaces de subdivision est tout d'abord la simplicité d'implémentation, mais aussi le fait que les schémas de subdivision peuvent s'appliquer sur des maillages de topologie arbitraire, et que la continuité de la surface peut être contrôlée localement comme on peut le voir sur la figure 2.7 (c) au niveau de la pointe qui ne s'arrondit pas comme les autres. Bien qu'il n'existe pas d'expression analytique directe pour de telles surfaces, elles sont très avantageuses au niveau de l'animation puisque l'on peut n'animer que le maillage grossier, les détails plus fins suivent au moment du rendu.

Sur la figure 2.5, on peut observer un personnage qui a été modélisé à l'aide de surfaces de subdivision et utilisé avec succès dans un court film d'animation de Pixar, *Geri's Game*.

2.1.4 Surfaces implicites

Une surface implicite est une surface définie par une fonction mathématique $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ qui associe une valeur scalaire à chaque point de l'espace. On peut imaginer un champ de potentiels. Ainsi, si l'on a une fonction $f(x, y, z)$, la surface sera définie par l'équation $f(x, y, z) = iso$.

Les fonctions implicites les plus connues et les plus simples sont les ellipsoïdes définis par une

fonction f de la forme :

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2}$$

Un avantage des surfaces implicites pour la modélisation est qu'il est facile d'obtenir un grand nombre de formes différentes à l'aide de fonctions mathématiques simples. Pour cela, elles peuvent être additionnées, soustraites, tordues comme le montre l'arbre de la figure 2.9.

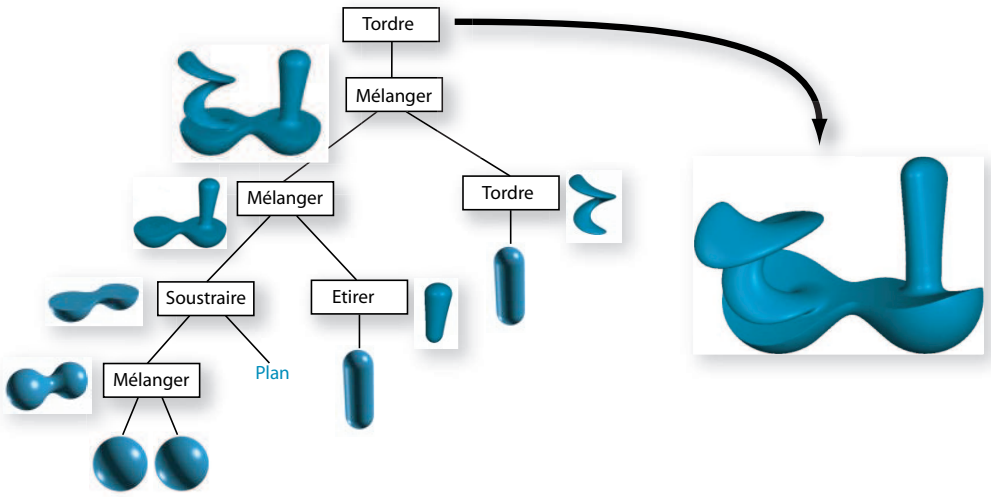


FIG. 2.9 – Arbre des actions possibles sur des primitives implicites pour obtenir un modèle détaillé (image extraite de [WGG99]).

Cependant, dans la pratique, le contrôle de la forme que l'on souhaite modéliser est assez difficile car il passe par l'utilisation de paramètres mathématiques peu explicites. D'où l'apparition et le succès des surfaces implicites à squelettes. Le principe est d'utiliser une primitive qui joue le rôle de squelette (un point, un segment, une courbe spline, une surface ...) pour laquelle il est possible d'évaluer la distance à un point donné et une fonction potentiel permettant de définir une *épaisseur* autour de ce squelette. La modélisation s'en trouve ainsi fortement simplifiée. De nombreux papiers se sont intéressés au choix de la fonction potentiel ainsi qu'au problème du mélange des potentiels ([Bli82, NHK +85, WMW86, Can93, HAC03]).

Une difficulté majeure de l'emploi de ces surfaces en animation est l'affichage à l'écran. Différents algorithmes existent pour créer une surface triangulée à partir de la surface implicite, mais ils sont tous très coûteux. On peut citer le lancer de rayons traditionnel comme étant le plus lent et les techniques de polygonalisation comme étant les plus répandues, notamment le *marching cubes* développé en 1987 par Bloomenthal [Blo87] d'une part et Lorensen et Cline [LC87] d'autre part. Plus récemment le *marching triangles* [HSIW96, AG01] a fait son apparition. Les autres techniques utilisées sont le rendu volumique et les méthodes par particules [BW90, WH94, DTC96].

Cependant, si ce type de surface est exploré dans le domaine de la recherche, son utilisation reste rare dans l'industrie, probablement à cause du manque de contrôle de l'infographiste sur la forme finale et de la lenteur de l'affichage. Comme le montrent les travaux de sculpture virtuelle de Ferley [FCG02] qui utilise une grille de potentiels, il est cependant possible d'obtenir des formes assez détaillées. Leclercq propose dans [LAG01] une méthode mixte, utilisable dans les jeux vidéos, mélangeant surfaces implicites et maillages polygonaux pour permettre de créer des formes plus *organiques* sans pour autant trop alourdir les temps de calcul (voir figure 2.10).

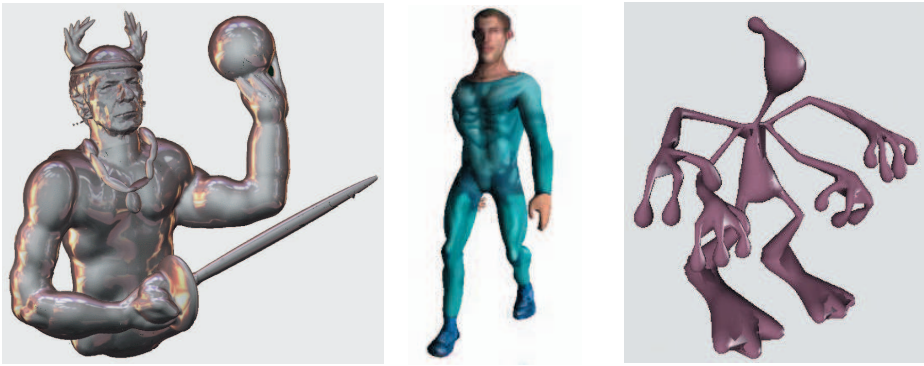


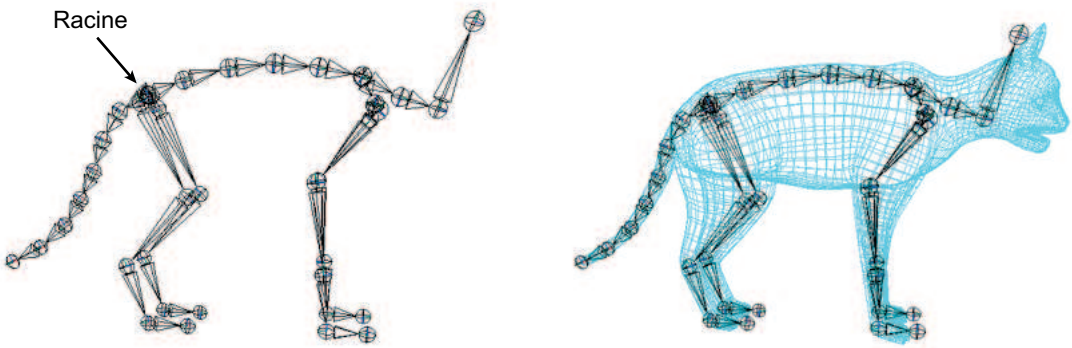
FIG. 2.10 – Exemples de modèles créés à l'aide de surfaces implicites (images extraites de [FCG02, LAG01, HAC03]).

En bref : Le type de modèle le plus couramment utilisé lors de l'animation reste la surface polygonale, tout simplement grâce à sa rapidité d'affichage. Les surfaces splines et plus particulièrement les NURBS sont cependant grandement utilisées pendant le processus de modélisation (elles sont le plus souvent converties en surfaces polygonales avant d'être utilisées pour l'animation) car le nombre de points de contrôle à manipuler est plus petit. Dans ces deux cas, le modèle résultant dépend entièrement de l'infographiste et de ses talents. Bien que les surfaces de subdivision et les surfaces implicites offrent une modélisation simplifiée pour l'infographiste, ceci se fait au détriment du contrôle et leur animation en temps-réel est souvent compromise en raison de temps d'affichage beaucoup trop longs. C'est pourquoi elles sont beaucoup moins utilisées dans l'industrie que dans la recherche.
Un nouveau type de représentation que nous n'avons pas évoqué plus haut est actuellement en plein essor. Il s'agit de la représentation par nuages de points. Le gros avantage par rapport aux autres types de surfaces est l'absence de connectivité entre les points et un rendu largement facilité par les cartes graphiques actuelles, mais cela rend leur utilisation plus difficile pour la détection de collisions par exemple. Néanmoins, des approches prometteuses ont récemment été proposées [KZ04].
Enfin, il est important de remarquer que la représentation utilisée pour modéliser l'objet n'est pas toujours celle qui est utilisée lors de l'animation pour le calcul des déformations. Il est parfois nécessaire d'utiliser des représentations volumiques de l'objet, lors de l'utilisation de modèles physiques notamment.

2.2 Animation des modèles structurés à squelettes

Nous avons présenté dans la section précédente différents types de modèles géométriques utilisables pour représenter l'apparence externe d'un objet, un personnage par exemple. La seconde étape lors de la mise en place d'une animation est d'associer une structure rigide, appelée *squelette*, à la géométrie du personnage pour lui permettre de se déformer. Grâce à l'utilisation de techniques de déformations présentées à la section suivante, l'animation du squelette engendre directement les déformations du personnage animé. Cette section s'attache à définir la notion de squelette puis à décrire les différents procédés existants pour l'animer.

2.2.1 Squelette et articulations



(a) (b)

FIG. 2.11 – Un exemple de squelette sur le modèle du chat : (a)-le squelette seul et sa hiérarchie ; (b)-le squelette dans le modèle géométrique.

Au sens de l'animation, un squelette est un ensemble de repères, organisés en hiérarchie, c'est à dire ayant des positions/orientations relatives les uns par rapport aux autres. Cela permet par exemple, quand on bouge le repère placé au niveau de l'épaule, de faire suivre automatiquement ceux placés au coude et au poignet. La base de la hiérarchie est la *racine*, tous les autres repères en dépendent. La figure 2.11 montre un tel squelette.

Dans la plupart des modèles simples, les repères du squelette sont placés aux *articulations idéales*. En effet, comme l'explique Maurel dans [Mau99], physiologiquement, il n'existe pas d'axe unique et fixe de rotation pour une articulation donnée (voir figure 2.12 (a)). Les forces s'appliquent en plusieurs points, les os *glissant* les uns sur les autres. Cependant, les translations étant la plupart du temps négligeables par rapport aux rotations, il est possible de modéliser une articulation par une rotule à 3 degrés de liberté. Dans certains cas, le modèle peut même être contraint à seulement 2 degrés de liberté, comme pour le genou par exemple (voir figure 2.12 (b)).

2.2.2 Cinématique directe

Lorsqu'un animateur anime *à la main* un squelette, il cherche à reproduire un mouvement observé ou une certaine déformation de l'enveloppe du personnage. Seuls le mouvement et ses effets sur le modèle sont importants, peu importe ce qui a généré ce mouvement. Ce processus est l'inverse de la réalité où le déplacement du squelette intervient en réponse à la contraction de certains muscles.

Pour mettre en place une animation, l'utilisateur spécifie un ensemble de positions clefs du squelette à des temps donnés. Pratiquement, cela consiste à affecter, à chaque repère, une position et une orientation dans l'espace. Ensuite, pendant l'animation, la position et l'orientation courantes de chaque repère au temps t sont interpolées à partir des poses spécifiées par l'infographiste.

Cette interpolation se matérialise pour chaque repère par un ensemble de courbes paramétriques $2D$, 3 pour la position en x, y et z, et 3 pour l'orientation. Les points de contrôle de ces courbes sont les positions/orientations définies par l'utilisateur à des temps donnés (voir figure 2.13). Chaque courbe correspond à la *trajectoire* d'un des paramètres du repère et son édition est aussi un moyen pour l'animateur de spécifier un mouvement plus fluide.

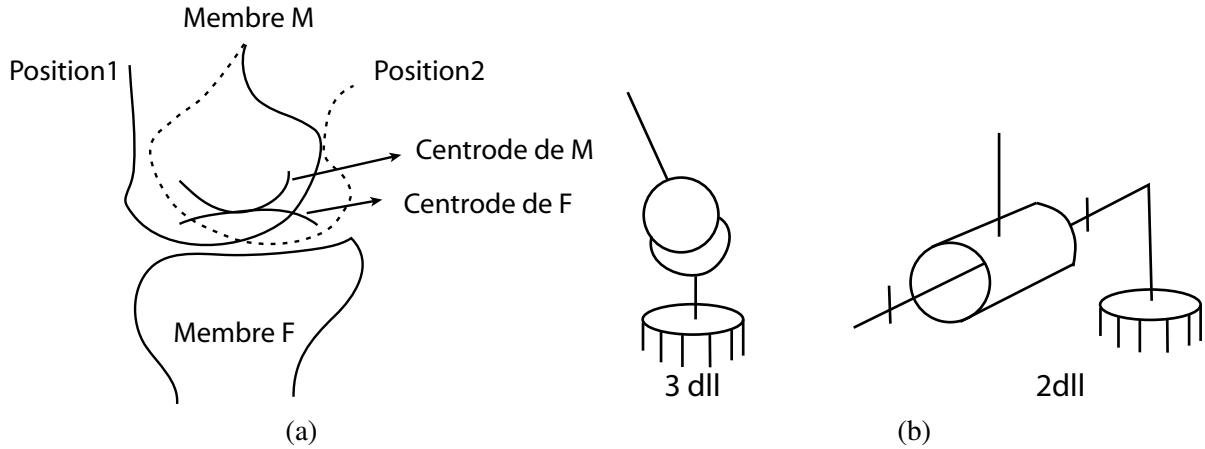


FIG. 2.12 – Articulations réelles et articulations idéales : (a)-lorsque deux membres d'une articulation réelle sont en mouvement relatif, les centrodes permettent de décrire les lieux des centres instantanés des deux objets et nous montrent le glissement qui intervient entre les deux membres (image extraite de [KG83]) ; (b)-en négligeant les translations, l'articulation peut être ramenée à une rotule (3 degrés de liberté) ou un modèle plus contraint (images extraites de [Mau99]).

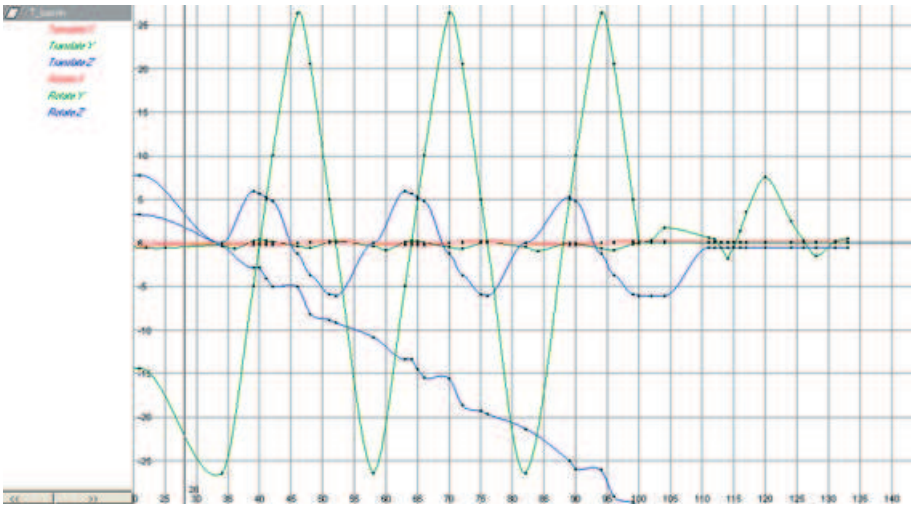


FIG. 2.13 – Exemple de trajectoires d'un repère : 3 courbes décrivent les translations en x, y, et z et 3 courbes décrivent les rotations selon les mêmes axes.

L'utilisateur a donc le choix, à tout moment, de déplacer directement un repère, ou bien d'éditer les courbes de trajectoires de chaque repère. Passer de l'un à l'autre est très souvent la solution retenue car elle permet de profiter des avantages des deux méthodes, l'édition de courbes aidant à créer des mouvements plus fluides et la manipulation des repères permettant de visualiser directement le résultat souhaité. Cependant, le gros inconvénient de la cinématique directe est la difficulté de positionner successivement tous les éléments d'une chaîne articulée en essayant d'atteindre une position/orientation précise pour le dernier élément. Cela peut se traduire par un manque de réalisme des animations créées, comme par exemple, le pied d'appui d'un personnage qui ne reste pas collé au sol pendant une marche.

2.2.3 Cinématique inverse

La cinématique inverse est donc très utilisée afin de calculer des poses clefs réalistes. C'est une technique issue de la robotique qui permet de calculer le positionnement des éléments d'une chaîne articulée en ne connaissant que la position de la base et celle du bout de la chaîne [GM85]. Il s'agit pour l'utilisateur de spécifier la position à atteindre par l'extrémité de la chaîne cinématique, les orientations des repères de la chaîne sont alors calculées automatiquement.

En animation, un mouvement peut être spécifié par une fonction continue dans le temps :

$$m(t) = (p(t), q_0(t), \ldots, q_n(t))$$

avec $p(t)$, la position globale du nœud racine, $q_0(t)$ son orientation, et $q_i(t)$ la transformation locale du repère i de la chaîne. La spécification de la position/orientation x_i d'un des repères de la chaîne articulée est une contrainte. Pour une pose donnée à l'instant t, l'ensemble des contraintes géométriques à atteindre est exprimé par $X = (x_0, \ldots, x_p)$. La cinématique inverse permet de calculer $m(t)$, telle que $X = F(m(t))$, F exprimant les propriétés cinématiques de la chaîne.

Ceci se fait en prenant en compte un certain nombre de contraintes au niveau des articulations (degrés de liberté, rotations maximales). Dans le cas général, la solution n'est pas unique et le problème se règle en ajoutant des contraintes de continuité ou de minimisation d'énergie [Arn94]. Des solutions analytiques efficaces ont été proposées pour des chaînes articulées telles que le bras dans [Pau81] et utilisées pour positionner les mains et les pieds de personnages [BKK +85, TGB00]. Des solveurs analytiques spécialisés ont été utilisés pour diverses applications, comme, par exemple, la marche [SM01].

Cependant, les solveurs analytiques ne sont pas adaptés pour les structures articulées générales. C'est pourquoi des approches numériques par recherche itérative de la meilleure solution sont souvent utilisées. De nombreuses techniques existent. Parmi elles, on retrouve celles qui utilisent la transposée de la matrice jacobienne impliquée lors de la résolution de l'équation précédente [WE84, SS88] au lieu de son inverse [MK85] ou bien encore qui cherchent à résoudre un problème de minimisation d'énergie qui permet de trouver une solution localement optimale satisfaisant un certain nombre de contraintes sur un certain nombre de repères [BMW87, ZB94].

Il est aussi possible de spécifier plusieurs tâches à atteindre, ce qui peut conduire à des contraintes conflictuelles. Dans ce cas, une solution est de pondérer les différentes tâches [BMW87, ZB94] ou bien d'utiliser un système de priorités [MK85, Bae01].

En définissant des contraintes différentes, ainsi que des priorités entre les contraintes, il est possible de générer des mouvements plus réalistes tout en laissant le soin à l'utilisateur de spécifier ces différentes contraintes et leurs priorités dans le cas de conflits [YN03, CB04]. Cela permet ainsi de faciliter la mise en place de l'animation tout en obtenant des résultats plus réalistes. Par exemple, l'utilisateur peut *coller* le pied du personnage au sol tout en l'obligeant à *saisir* un objet avec la

main. Les positions/orientations du genou, de la cheville, de l'épaule et du coude sont alors calculées automatiquement pour satisfaire au mieux les contraintes.

2.2.4 Animation dynamique

Contrairement à l'animation par cinématique (directe ou inverse), l'animation dynamique permet de calculer un mouvement au cours du temps qui dépend des lois de la mécanique et qui prend en compte les causes qui ont généré ce mouvement. On parle souvent de *modèle générateur* dans ce cas. Ainsi, la position, la vitesse, l'accélération des différents repères du squelette sont directement liés aux forces appliquées sur celui-ci. Ceci est effectué en prenant en compte un ensemble de forces extérieures (gravité, vent) et des forces intérieures (générées par les muscles par exemple) et en appliquant les lois de la dynamique des corps rigides. L'avantage est bien sûr la possibilité de générer des mouvements plus réalistes, plus nombreux. Mais cela se paie par un manque de contrôle sur le mouvement généré et des équations souvent difficiles et longues à résoudre. Cependant, Hodgins et al. [HWBO95, Hod96] sont parvenus à générer des mouvements complexes de course et de saut en prenant en compte des données réalistes de masse et d'inertie, tirées de la littérature biomécanique.

2.2.5 Capture du mouvement

Afin de créer des animations encore plus réalistes, l'animateur peut avoir recours à de la capture du mouvement.

À l'aide de capteurs

Pour cela, un acteur doit produire le mouvement désiré par l'animateur. Il est équipé de quelques capteurs principalement placés au niveau des articulations (des capteurs infrarouges par exemple) et les trajectoires de ces capteurs sont capturées, à l'aide de plusieurs caméras disposées tout autour de lui, dans le cas d'un système optique par exemple. Ainsi, par un suivi de la position des marqueurs au cours du temps et dans l'espace, il est possible de reconstruire le mouvement de l'acteur et de l'appliquer à un squelette.

Cette manipulation n'est cependant pas immédiate, car il faut faire face à certains problèmes comme l'occlusion des capteurs ou le bruit récupéré sur le mouvement. Un autre problème de cette technique est la spécificité du mouvement capturé qui n'est applicable qu'à un squelette de même taille. De nombreux papiers [Gle98, MBBT00] traitent ce problème en adaptant automatiquement les paramètres de l'animation récupérée par la capture à la morphologie du personnage que l'artiste souhaite animer. On peut aussi citer la technique des *motion graphs* [KGP02] ou les techniques de pondération [KG04] qui permettent de synthétiser des mouvements réalistes à partir de bases de données de mouvements capturés.

Néanmoins, l'appareillage reste quelque peu compliqué et cher. De plus, si capturer des mouvements humains est aisé, il n'en est pas de même lorsqu'il s'agit d'animaux, volumineux ou sauvages !

À partir d'une vidéo

C'est pourquoi de nouvelles approches très originales tentent de capturer des mouvements à partir de vidéos [FRDC04]. La technique permet de détecter les images clefs dans la vidéo et le travail de l'infographiste se limite donc à la mise en place de 4 ou 5 poses du modèle $3D$ au lieu de devoir créer

l'animation complète. C'est aussi un excellent moyen pour essayer de réappliquer à des personnages des animations non réalistes de dessins animés [BLCD02, dJB04].

En bref : L'animation du squelette, constitué d'une hiérarchie de repères, s'effectue en modifiant les positions et orientations des repères. En utilisant la cinématique directe, l'animateur reste maître du mouvement et, par conséquent, les résultats obtenus dépendent fortement de l'animateur. La cinématique inverse permet de composer des mouvements plus réalistes mais le rôle de l'utilisateur reste très important. L'utilisation de la dynamique pour créer des animations réalistes donne de bons résultats, mais parfois difficiles à contrôler. C'est pourquoi les techniques de capture du mouvement, soit à l'aide de capteurs, soit à partir de vidéos sont en plein essor. Elle permettent de plus de constituer des bases de données de mouvements qui peuvent ensuite être utilisées par des applications de plus haut niveau, comme par exemple le contrôle de l'animation par des scénarios issus d'analyse comportementale.

2.3 Techniques de déformation

Disposer d'un squelette d'animation et d'une surface représentant l'objet ne suffit pas à créer une animation. En effet, il est indispensable d'utiliser des outils permettant de spécifier comment la géométrie de l'objet se déforme par rapport au mouvement du squelette interne. La déformation d'un maillage géométrique passe par l'utilisation de techniques appropriées qui peuvent être géométriques ou physiques. Comme nous allons le voir au cours cette section, il est parfois nécessaire de passer à une représentation volumique de l'objet afin de calculer les déformations souhaitées.

2.3.1 Techniques géométriques

Nous avons choisi de présenter dans cette section trois techniques géométriques de déformation de maillages. Ce sont toutes des techniques très répandues et nous verrons au chapitre 3 comment elles peuvent être utilisées dans le cadre de l'animation d'objets structurés tels que des personnages.

2.3.1.1 Interpolation de formes clefs

Cette technique est sans doute la plus utilisée en animation faciale, car elle peut s'appliquer à un maillage sans squelette interne. Nous allons donc l'expliquer en prenant comme exemple l'animation faciale, dont les paramètres sont plus faciles à décrire, mais la même technique peut bien sûr être utilisée pour l'animation de muscles [LN98] ou d'un objet quelconque.

L'utilisateur doit spécifier des formes clefs correspondant à différentes expressions : la tristesse, la joie, la colère (ou des formes extrêmes pour différentes positions articulaires). Ensuite, la position des sommets du maillage est obtenue par combinaison linéaire de la position des sommets dans les maillages représentant les expressions extrêmes. Par exemple, si le personnage est *à moitié en colère*, la position du sommet i sera donnée par la moitié de la position du sommet dans le maillage *neutre* plus la moitié de la position du sommet dans la position *en colère*. Ces formes sont ensuite combinées pour obtenir la forme finale pour les positions intermédiaires.

Mathématiquement, si n est le nombre de formes clefs, S_i la forme clef représentant la i_{eme}

expression et w_i, le poids de l'expression S_i, la forme finale S_f est égale à :

$$S_f = \sum_{i=0}^{n} w_i S_i \tag{2.1}$$

Les problèmes liés à cette approche sont divers. Tout d'abord, les différentes formes clefs ne sont pas indépendantes : l'artiste doit dessiner des formes qui n'entrent pas en conflit les unes avec les autres. En effet, les formes sont ajoutées les unes aux autres, et rien ne garantit que l'effet de l'une ne soit pas effacé par l'autre, ou, au contraire, trop renforcé, surtout dans le cas où plus de deux formes clefs entrent en jeu dans le calcul de la forme courante. Ensuite, les formes clefs sont différentes d'un personnage à l'autre. Il n'est donc pas possible de réutiliser les formes créées pour un personnage donné sur un autre personnage. Enfin, dessiner un certain nombre de formes pour différentes poses est une tâche fastidieuse pour l'artiste, mais d'un autre côté, il reste relativement maître du résultat.

2.3.1.2 Déformations dans les sous-espaces du squelette

Le *skinning* ou *Skeleton Subspace Deformations* [LCF00, Blo02] est une technique d'animation pour laquelle les sommets de la surface d'un objet sont déplacés en réponse au mouvement d'un squelette interne. Le squelette est composé d'une hiérarchie de repères comme expliqué à la section 2.2.1 et recouvert d'un maillage appelé la *peau*. Les repères du squelette sont bougés, soit par cinématique directe par l'animateur, soit par cinématique inverse ou capture du mouvement pour produire le mouvement, entraînant les sommets de la peau avec eux.

Pour cela, chaque sommet du maillage doit dépendre d'au moins un repère. Dans le cas le plus simple (dépendance à un seul repère), on pourra considérer que tous les sommets du maillage compris entre le repère représentant le bassin et le repère représentant le genou appartiennent à la cuisse et que leur mouvement dépend donc du bassin. C'est la forme la plus simple de *skinning*, appelée *skinning rigide*, où chaque sommet de la peau subit la même transformation rigide que le repère du squelette auquel il est rattaché. Cependant, cela conduit à des déformations non réalistes qui se traduisent par des étirements ou des intersections du maillage autour des articulations, ou parfois même des trous lorsque le maillage a été préalablement découpé en *membres* (voir figure 2.14).

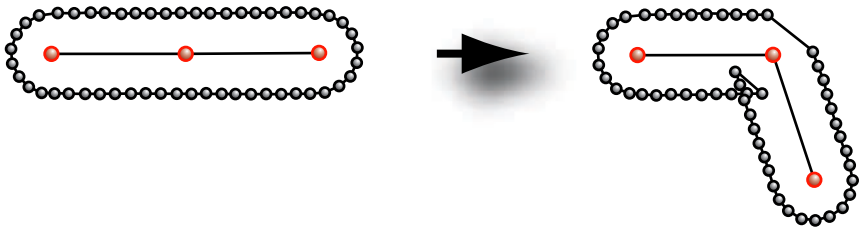


FIG. 2.14 – Illustration de la technique du *skinning rigide* : le maillage est étiré d'un côté de l'articulation et il s'intersecte de l'autre côté (image extraite de [Blo02]).

Une solution pour éviter les problèmes autour des articulations est d'utiliser un *smooth skinning*, que nous désignerons par *skinning* dans tout le reste de ce document, qui autorise un sommet du maillage à être influencé par plusieurs repères. Dans ce cas, la position d'un sommet i est définie comme étant une somme pondérée de ses positions qu'il aurait dans les différents repères dont il dépend, selon l'influence qu'il subit de tel ou tel repère. La position finale p_i d'un point i de la surface se trouve donc dans le sous-espace défini par les transformations rigides de ce point par un

certain nombre n de repères :

$$p_i = (\sum_{f=0}^{n} w_{if} T_f)\nu_i \tag{2.2}$$

avec :
ν_i, la position du sommet i avant transformation (position neutre) ;
T_f, la transformée du repère f ;
w_{if}, le poids associé au point i qui exprime la dépendance de i au repère f.

Les poids w_{if} jouent un rôle essentiel au niveau de l'apparence visuelle du résultat. Dans une approche classique ils sont peints *à la main* par l'infographiste sur la surface du modèle, ce qui s'avère être long et demande énormément de savoir-faire. Afin de les calculer automatiquement, il est néanmoins possible d'utiliser une information de distance os/sommet dans la position de repos. Afin de calculer des distances adéquates, nous considérons par exemple que l'*os* est un repère défini comme le milieu de deux repères articulations. Soit b le repère d'un tel os. Par exemple, le poids w_{if} du sommet i dans le repère f pourra être calculé de la manière suivante : $w_{if} = \frac{1}{d_i^2}$, d_i représentant la distance entre le sommet i et le repère b de l'os. En faisant cette opération pour tous les sommets et en normalisant les poids obtenus, on obtient des poids raisonnables qui varient sans discontinuité le long du maillage, tant que les parties du maillage considérées sont simples (comme un bras, une jambe). Cette technique ne fonctionne plus du tout au niveau du torse et des épaules par exemple. Nous verrons au chapitre 3 comment certaines approches cherchent à calculer des poids plus complexes et mieux adaptés [Blo02].

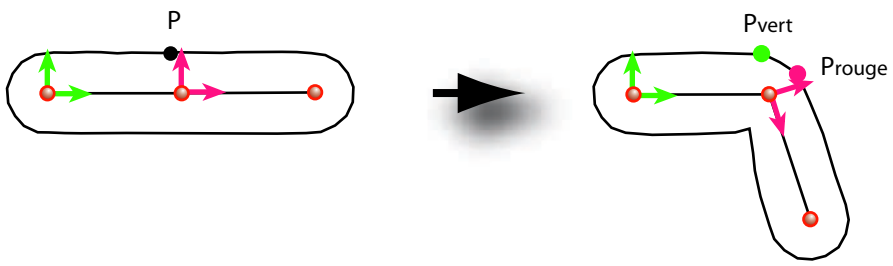
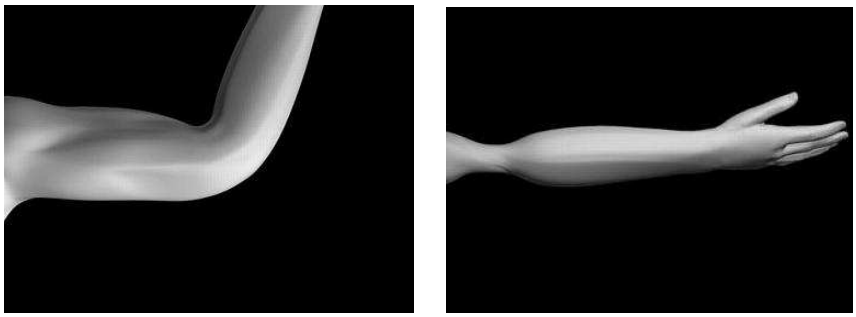


FIG. 2.15 – Illustration de la technique du *skinning* utilisant deux repères : le point P se déplace en un nouveau point P qui se trouve obligatoirement sur la droite reliant P_{vert} à P_{rouge}.

Quelle que soit la méthode utilisée pour le calcul des poids, le principe reste le même. Malheureusement, cette technique souffre de plusieurs problèmes. Tout d'abord, la déformation est restreinte au sous-espace associé. Dans le cas où un point dépend de deux repères, sa position finale se situe obligatoirement sur le segment reliant les deux points représentant sa position dans chacun des repères (voir figure 2.15). C'est à dire que les sommets ne peuvent pas toujours se retrouver là où l'on voudrait qu'ils soient. En particulier, cette technique est inadaptée pour le coude ou l'épaule pour lesquels la déformation recherchée ne se trouve pas dans le sous-espace. En pratique, cela se traduit par un plat qui apparaît là où l'os du coude devrait se trouver (voir figure 2.16 (a)). D'autre part, en comparaison avec l'interpolation de formes clefs, cette technique ne permet pas les manipulations directes. En effet, pour chaque point de la surface, il y a un poids par repère du squelette qui l'affecte. Il faut donc éditer les maillages des poids w_{if} pour corriger une déformation manuellement, un travail fastidieux qui ne permet pas toujours d'obtenir le résultat voulu.

Cependant, malgré ces quelques inconvénients, la technique du *skinning* reste la plus utilisée lorsqu'il s'agit d'animer des personnages articulés. L'amélioration automatique des poids [Blo02]



(a) (b)

FIG. 2.16 – Problèmes liés à la technique du *skinning* : (a)-aplatissement au niveau du coude ; (b)-amincissement au niveau de l'articulation quand le bras est vrillé.

ou l'utilisation du *skinning* en combinaison avec d'autres techniques [LCF00, KJP02] permettent de corriger certains problèmes, sans toutefois apporter une solution pleinement satisfaisante.

2.3.1.3 Déformations de formes libres

Une autre méthode largement répandue pour l'animation d'objets déformables est la technique des déformations de forme libre, plus connue sous le nom de FFD, *Free Form Deformations* en anglais. Elle a été proposée pour la première fois en infographie par Sederberg et Parry [SP86] en 1986.

Le principe de la technique est simple : l'objet que l'on souhaite déformer est plongé dans la grille de contrôle $3D$ d'un volume spline tridimensionnel de façon à ce que la position de chaque point de l'objet dans la grille puisse être définie à l'aide d'une paramétrisation unique. Ainsi, pour chaque point de l'objet, ses coordonnées paramétriques sont déterminées au repos et stockées. Ensuite, la grille est déformée, et, puisque les points gardent les mêmes coordonnées, l'objet se trouve déformé lui aussi.

Dans la méthode proposée par Sederberg et Parry [SP86], cette grille de contrôle définit un volume de Bézier et sa déformation est contrôlée par les points de contrôle traditionnels[4]. L'intérêt d'interpoler les positions des points de l'objet déformé par des fonctions tri-paramétriques est de pouvoir générer des déformations lisses, la continuité de la surface étant préservée. Cette technique permet ainsi des déformations globales de l'objet, la continuité de la surface déformée n'étant cependant pas assurée dans le cas d'une déformation locale (si seulement une partie de l'objet est plongée dans la grille de contrôle). Griessmair et Purgathofer [GP89] reprennent le même principe mais en utilisant des fonctions de base B-splines tri-paramétriques.

La force des FFD est qu'elles peuvent être appliquées à différentes représentations graphiques : points, polygones, splines, surfaces paramétriques et surfaces implicites. Cependant, le gros inconvénient des premières approches proposées est que la grille de contrôle est parallélépipède pour simplifier la paramétrisation des points. C'est en effet la partie la plus délicate de l'algorithme et c'est pourquoi les approches suivantes se sont attachées à diversifier la forme de la grille de contrôle utilisée.

[4]Faire l'analogie avec les courbes et les surfaces splines présentées à la section 2.1.2, en se plaçant dans un espace de dimension 3 et en considérant les polynômes de Bernstein comme fonctions de base.

Extension à des grilles non parallélépipèdes

Coquillart [Coq90] s'attaque donc en premier à l'utilisation de grilles non parallélépipèdes. Sa technique, appelée *Extended Free Form Deformation* utilise des grilles entre autres cylindriques obtenues par combinaison de plusieurs grilles parallélépipèdes et de ce fait doit utiliser des méthodes numériques pour calculer les coordonnées des points.

Dans [LCJ93], Lazarus propose des déformations *axiales*. Le principe est le même que pour les FFD à la différence que la grille de contrôle n'est plus un cube, mais un axe (une courbe de Bézier par exemple). Les points de l'objet sont attachés à l'axe et paramétrés suivant cette attache. Ainsi, lorsque l'axe se déforme, l'objet se retrouve déformé lui aussi. L'axe peut se situer à l'extérieur ou à l'intérieur de l'objet. Encore une fois, c'est le choix de la paramétrisation et le calcul des coordonnées des points qui sont les étapes difficiles de l'algorithme.

Plus récemment, MacCraken [MJ96] propose une nouvelle technique de FFD qui permet d'utiliser une grille $3D$ de topologie arbitraire, c'est à dire contenant des cellules qui n'ont pas forcément 6 faces. Le schéma de subdivision de Catmull-Clark est utilisé pour raffiner cette grille $3D$ en un ensemble de grilles plus petites qui, dès la première subdivision, peuvent toutes se ramener à un ensemble de tétraèdres dans lesquels il est aisé de calculer les coordonnées d'un point.

FFD animées

L'autre évolution des FFD se place au niveau de l'animation. Ainsi, Coquillart [CJ91] anime des objets en créant deux grilles de contrôle, une initiale et une finale, et en interpolant les points de contrôle entre la grille initiale et la grille finale.

Chadwick [CHP89] puis Faloutsos [FvdPT97] rajoutent de la dynamique à la déformation en ajoutant des masses ponctuelles ainsi que des paramètres d'élasticité à l'objet. Faloutsos [FvdPT97] fait ensuite vibrer l'objet à l'aide de modes définis par l'utilisateur. Dans [FvdPT97], l'utilisation de grilles hiérarchiques permet de définir des déformations globales, mais aussi locales de l'objet, sans toutefois garantir la continuité C^1 de la surface de l'objet.

Ces deux derniers modèles font appel à des propriétés physiques, qui, mixées avec une méthode géométrique, permettent d'obtenir des comportements dynamiques de l'objet simulé. Nous allons voir plus en détail à la section suivante quels sont les modèles purement physiques qui existent et en quoi leur utilisation en infographie est particulièrement intéressante.

2.3.2 Modèles physiques

Les modèles physiques, contrairement aux modèles géométriques, s'attachent à décrire les causes des déformations plutôt que de spécifier la déformation en elle-même. L'intervention de l'animateur est dans ce cas limitée au réglage des paramètres de la simulation qui sont les paramètres physiques des objets et les conditions initiales. Les motivations pour l'utilisation de tels modèles résident tout d'abord dans la recherche d'un réalisme poussé de la déformation. En effet, les calculs s'effectuent à partir des lois de la physique et des propriétés intrinsèques du matériau simulé telles que sa masse, sa raideur ou sa limite élastique. De plus, le rôle de l'utilisateur étant réduit, il devient aisé de gérer un grand nombre d'objets similaires.

2.3.2.1 Quelques éléments de physique

Les objets déformables, tels que les tissus ou les organes humains ou animaux, peuvent être décrits par la mécanique des milieux continus. Pour étudier les déformations d'un milieu continu donné, il

est nécessaire de définir un état de référence pour l'objet, ou *état de repos*. Le mouvement d'un objet déformable sera alors défini comme étant *une suite continue de déformations entre la configuration de référence et les configurations déformées au cours du temps*.

Pour décrire ces déformations, il existe deux formalismes en mécanique des milieux continus : la description lagrangienne et la description eulérienne. Dans le premier cas, l'objet est modélisé par ses particules i suivies dans le temps alors que dans le second cas, c'est le milieu qui est décrit par son champ spatial des vitesses défini en chaque point M de l'espace.

La description lagrangienne du mouvement d'un corps déformable consiste à se donner à chaque instant t, l'expression des vecteurs positions $\vec{x}_i$ pour chaque point matériel i de l'objet par rapport à sa position dans l'état de référence. Cette description donne donc une fonction position valable en chaque point du solide qui décrit de façon continue la déformation. Elle est généralement la mieux adaptée lorsqu'il s'agit de solides déformables dont la structure ne change pas, c'est à dire lorsque les points matériels le constituant gardent toujours le même voisinage au cours du temps.

La description eulérienne du mouvement d'un corps déformable consiste à observer, en chaque point M du milieu et à chaque instant t, l'évolution des paramètres. Cette description donne un champ de vitesses $\vec{v}$ en chaque point M de l'espace. Elle est le plus souvent utilisée en mécanique des fluides puisque les particules ont tendance à ne pas garder leur structure et que la trajectoire d'un atome d'eau dans un fleuve par exemple, a beaucoup moins de signification que la vitesse d'écoulement de l'eau en tel ou tel point de la rivière.

Déformations physiques

Les déformations d'un solide sont soumises à des lois de comportement. Suivant la nature et l'intensité des forces qui s'appliquent sur l'objet déformable, son comportement sera différent. Lorsque les forces sont peu importantes, la déformation de l'objet est entièrement élastique. La viscosité peut alors être prise en compte si l'on s'intéresse à la dynamique de la déformation. Lorsque les efforts appliqués sont plus importants, la déformation devient plastique et l'objet peut aller jusqu'à se fracturer si certains seuils sont dépassés (début de la zone de striction, voir figure 2.17).

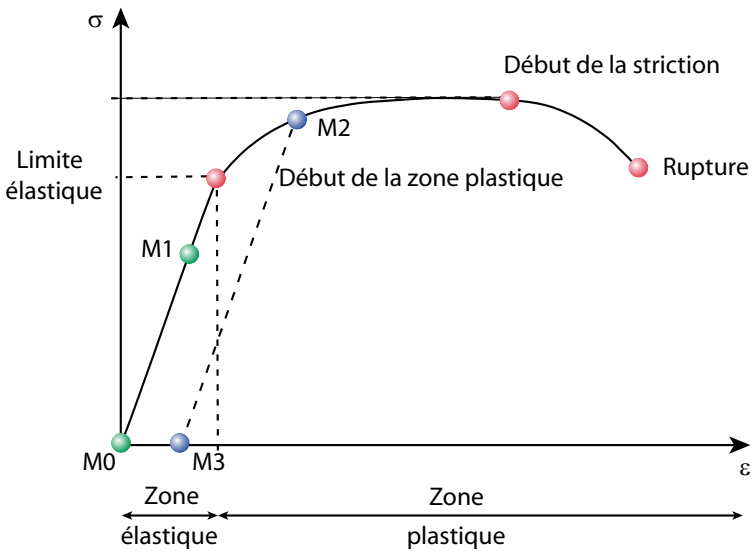


FIG. 2.17 – Loi de comportement d'un matériau élastique qui trace l'évolution des déformations (en abscisse) en fonction des contraintes appliquées à l'objet (en ordonnée).

Élasticité L'élasticité est la capacité d'un corps à s'étirer ou à se comprimer et à revenir à sa position initiale lorsque les efforts sont relâchés. On peut alors comparer l'objet à un ressort parfait qui stocke de l'énergie lorsqu'il est comprimé ou étiré et qui la restitue intégralement lorsque la force cesse de s'appliquer (ainsi, le point étiré $M1$ de la figure 2.17 revient en $M0$ lorsque les efforts sont relâchés). Cette déformation est complètement réversible tant que les forces appliquées restent faibles. Par contre, si la force appliquée à l'objet est au dessus d'un certain seuil (la limite élastique), on sort du domaine élastique de l'objet pour entrer dans son domaine plastique. Les déformations sont alors partiellement irréversibles. Dans le domaine élastique, la courbe contraintes/déformations est une droite dont la pente est le module d'Young E, caractéristique du matériau (voir figure 2.17).

Viscosité La viscosité intervient lorsque la vitesse de déformation fait intervenir une force supplémentaire qui s'oppose à la déformation. C'est le cas de la plupart des matériaux. Elle agit un peu comme un frein à la déformation en *retardant* le retour à l'équilibre de l'objet. Son effet n'a donc d'intérêt que si l'on s'intéresse à la *dynamique* de la déformation.

Plasticité Lorsque le matériau entre dans son domaine plastique, la force appliquée à l'objet n'est plus entièrement restituée. Il y a des pertes d'énergie qui, concrètement, rendent la déformation irréversible. Ainsi, lorsque les contraintes sont relâchées, le matériau revient à un état d'équilibre différent de l'état de repos initial. Sur la courbe de la figure 2.17, $M2$ se situe dans la zone plastique. Quand les forces cessent de s'appliquer, le matériau revient dans la position $M3$ et non pas $M0$.

Modèle continu

Les équations du mouvement d'un objet déformable s'expriment de la manière suivante :

$$div\sigma + \rho(F - \gamma) = 0 \tag{2.3}$$

avec :
div, l'opérateur de divergence ;
F, la densité de force par unité de masse appliquée sur la particule d'intérêt ;
γ, l'accélération ;
σ, le tenseur des contraintes ;
et ρ, la masse volumique.

Si l'on se place dans le cas d'un matériau élastique isotrope, sa loi de comportement peut être exprimée par la loi de Hooke :

$$\sigma = A\epsilon \tag{2.4}$$

avec :
ϵ, le tenseur des déformations ;
et A, la loi de comportement du matériau.

A est une matrice décrivant la loi de comportement du matériau à partir des coefficients de Lamé μ et λ ou du module d'Young E et du coefficient de Poisson ν. Ces deux jeux de coefficients sont équivalents.

Calculer les déformations de l'objet, c'est, en formalisme lagrangien, exprimer $\vec{u}$, champ de déplacement pour l'objet, qui dépend du tenseur des déformations ϵ : $\epsilon = \epsilon(\vec{u})$. Par exemple, si l'on se place en hypothèse des petites perturbations (HPP), le tenseur des déformations ϵ est relié au champ de déplacement $\vec{u}$ par l'équation suivante :

$$\epsilon = \frac{1}{2}(\nabla\vec{u} + \nabla\vec{u}^T) \tag{2.5}$$

L'équation 2.3 est continue, c'est à dire valable en tout point de l'objet. Pour la résoudre, il faudrait en théorie intégrer cette équation sur tout le volume de l'objet, ce qui est impossible. C'est pourquoi en pratique, il est courant d'effectuer une discrétisation spatiale de l'objet, soit en masses et ressorts (modèle discret), soit en éléments finis (modèle continu). Cependant, des techniques innovantes telles que les splines dynamiques [RNG99, RNN00] tentent d'intégrer toutes les grandeurs physiques continûment, ce qui n'est pas le cas pour les éléments finis où les forces par exemple restent concentrées aux nœuds.

2.3.2.2 Les différents modèles

Dans cette section, nous nous intéressons aux techniques qui discrétisent le problème en espace afin de résoudre l'équation 2.3. Nous allons successivement présenter trois modèles permettant d'approcher les déformations de manière plus ou moins exacte : les réseaux masses-ressorts, les éléments finis et la synthèse modale.

Modèles masses-ressorts : déformations nodales

Le modèle le plus simple à comprendre et à utiliser est le modèle masses-ressorts. Il s'agit de discrétiser un volume $3D$ en un ensemble de volumes élémentaires. Ces volumes peuvent être des cubes, des tétraèdres par exemple. La masse totale de l'objet est ensuite concentrée aux sommets des volumes élémentaires et l'élasticité du matériau est approchée par des ressorts placés sur les arêtes. Cette discrétisation conduit à une approximation de la déformation qui est généralement très grossière et qui souvent, même dans des cas simples, ne permet pas d'obtenir un comportement *réaliste* du matériau déformé. Néanmoins, les calculs mis en jeu sont simples et rapides, ce qui peut s'avérer avantageux dans le cadre d'applications temps-réel où les temps d'éxécution priment sur l'exactitude des résultats.

Lors de l'animation, il s'agit de calculer le déplacement de chaque masse pour obtenir le maillage déformé au cours du temps. Pour cela, il suffit d'appliquer sur chaque masse le principe fondamental de la dynamique ou seconde loi de Newton qui dit que la somme des forces s'exerçant sur une masse ponctuelle est égale au produit de la masse m par l'accélération γ :

$$F = m\gamma \tag{2.6}$$

Un ressort reliant deux masses crée une force F sur chacune des masses égale à :

$$F = -ku \tag{2.7}$$

avec :
k, la raideur du ressort ;
et u, son élongation.

En appliquant le principe de la dynamique (équation (2.6)) à chaque masse m, on en déduit :

$$m\gamma + ku = 0$$

Lorsque l'on prend en compte la viscosité (dans le cas d'une simulation dynamique), il faut ajouter un coefficient d'amortissement a qui agit sur la vitesse d'élongation du ressort. La force F s'écrit alors :

$$F = -ku + av$$

avec :
a, le coefficient d'amortissement ;
et v, la dérivée de u par rapport au temps.

Pour obtenir le déplacement de chaque masse, il suffit de résoudre cette équation pour chaque masse, en prenant en compte toutes les forces qui agissent sur cette masse, provoquées par les ressorts qui y sont attachés, mais aussi la gravité par exemple. Cette résolution se fait au cours du temps par un schéma d'intégration numérique tel que ceux présentés à la section 2.3.2.3.

Modèles continus : éléments finis

Le principe des éléments finis repose, comme pour les masses-ressorts, sur une discrétisation de l'espace en volumes élémentaires reliés entre eux aux nœuds a_i. Seulement, à la différence des masses-ressorts, au lieu de concentrer les masses et les raideurs aux nœuds, elles sont définies à l'aide de fonctions de base qui permettent d'interpoler la masse et la raideur en tout point du volume élémentaire. La base de fonctions doit être choisie judicieusement en accord avec le type de déformations que l'on souhaite observer, puisque les déplacements doivent pouvoir être représentés par une combinaison linéaire des fonctions de base. En $3D$, ces fonctions sont utilisées trois fois, pour x, y, et z. La projection du déplacement u sur la base s'écrit alors :

$$u = \sum_i a_i \Phi_i(M) \tag{2.8}$$

avec :
a_i, les nœuds du maillage volumique ;
et Φ_i, les fonctions d'interpolation.

Il faut noter que le maillage est découpé en volumes élémentaires de façon à prendre en compte certaines propriétés de masse ou d'élasticité de l'objet et qu'il peut être uniforme ou non. Le plus souvent, il est construit en prenant des éléments plus nombreux et plus petits dans les parties du modèle où l'on souhaite voir plus de détails.

Nous décrivons le déplacement u par un ensemble de fonctions de bases. Cependant, cette approximation engendre une erreur par rapport à la loi de comportement (équation (2.4)) réelle du matériau.

En minimisant cette erreur, il est possible, par une série de calculs que nous ne détaillerons pas ici, d'obtenir l'équation d'équilibre suivante :

$$F = Ku \tag{2.9}$$

avec :
K, la matrice de raideur $[n * n]$ dans la base des Φ_i et qui dépend de A et ϵ ;
F, le vecteur des forces extérieures $[n * 1]$ dans la base des Φ_i ;
et u, le vecteur de déplacement que l'on recherche $[n * 1]$.

Sous cette forme simplifiée, le corps peut être vu comme un énorme ressort $3D$ de raideur K qui retourne une force F.

Si l'on s'intéresse à la dynamique du système, on peut alors ajouter de la viscosité qui agit sur les vitesses. Elle est prise en compte au niveau de la loi de comportement (dans la matrice A). Dans ce cas, l'accélération de l'équation 2.3 n'est plus nulle. On obtient alors

$$F = M\gamma + Dv + Ku \tag{2.10}$$

avec :
M, matrice de masse des éléments ;

$\gamma = \ddot{u}$, accélérations des masses ;
D, matrice d'amortissement ;
et $v = \dot{u}$, vitesses des masses.

Les équations (2.9) et (2.10) seront rencontrées au chapitre 3 pour décrire la statique et la dynamique des systèmes d'éléments finis.

Modèle vibratoire : déformations globales

L'analyse modale a été utilisée pour la première fois en infographie par Pentland et Williams [PW89] en 1989. Elle permet de décrire, de façon simple, les déformations dynamiques d'un objet déformable. Cette technique est une simplification des éléments finis. Elle consiste à projeter les déformations possibles sur une base de fonctions réduite. On cherche donc à calculer l'intensité des déformations considérées pour pourvoir ensuite les recombiner pour créer les déformations pendant le mouvement.

Pour cela, on considère l'équation (2.10). M et K étant symétriques positives, on les diagonalise dans un même repère orthogonal (en cherchant les valeurs et vecteurs propres de $M^{-1}K$). Afin de diagonaliser l'amortissement D dans le même repère que les deux autres matrices, on choisit $D = \alpha M + \beta K$.

L'équation (2.10) peut alors s'écrire sous la forme de n équations :

$$\bar{f}_i = m_i \ddot{\bar{u}}_i + k_i \bar{u}_i$$

La résolution se déroule donc de la manière suivante :

1. On résout $M^{-1}K\Phi = \lambda\Phi$ en recherchant les vecteurs propres Φ_i et leurs valeurs propres λ_i associées. On obtient donc des couples (λ_1, Φ_1), (λ_2, Φ_2), ... qui vérifient

$$M^{-1}K\vec{\Phi}_i = \lambda_i \vec{\Phi}_i$$

   λ_i est alors la fréquence de vibration et Φ_i la forme de déformation associée.

2. On a donc un système de n équations indépendantes à n inconnues facile à résoudre :

$$\bar{F} = \bar{M}\ddot{\bar{u}} + \bar{D}\dot{\bar{u}} + \bar{K}\bar{u}$$

   Chaque équation décrit un mode vibratoire de l'objet. D'autre part, on ne résout que les modes qui nous intéressent. Si seuls les deux premiers modes nous intéressent, nous ne résolvons que deux équations (les modes étant trouvés directement par ordre d'importance, i.e., l'effet du premier mode est plus visible que celui du 2 ème, etc ...).

3. Pour créer la simulation, il suffit ensuite de faire une superposition linéaire des modes retenus.

Dans la pratique, il n'est pas nécessaire de calculer ces modes sur l'objet réel, mais on peut le faire sur un objet de forme simple (pavé par exemple) et l'appliquer ensuite au modèle déformable. D'autre part, une fois les modes (fréquence et forme) pré-calculés, leur utilisation se fait en temps-réel lorsque peu de modes sont utilisés puisqu'il s'agit d'effectuer une simple combinaison linéaire des modes utilisés. Cette technique de calcul du déplacement est adaptée dans le cas de systèmes à faible degré de liberté et lorsque le mouvement que l'on veut spécifier peut se satisfaire de peu de *modes*. Cet algorithme est particulièrement bien adapté pour faire du niveau de détail puisque l'on ajoute les hautes fréquences seulement si elles sont nécessaires et si le système de visualisation le permet.

2.3.2.3 Intégration numérique

Un modèle générateur spécifiant la dynamique d'un système mécanique s'exprime, à chaque image, par la résolution d'un système d'équations différentielles du second ordre non linéaires. Ces équations sont résolues par l'emploi d'une méthode d'intégration numérique adaptée. Le choix de cette méthode se fait en fonction du conditionnement des équations que l'on souhaite résoudre pour obtenir un maximum de stabilité dans la résolution. Cette section présente les schémas d'intégration explicites et implicites les plus courants. Pour plus de détails, le lecteur pourra par exemple se reporter à [Noc01]. Dans la suite, le pas de temps sera exprimé par dt.

Les méthodes explicites

Les méthodes explicites calculent les valeurs de la position et de la vitesse d'un point uniquement à partir des valeurs connues (aux pas de temps d'avant). Parmi les méthodes existantes, la plus simple est la méthode d'Euler. C'est une méthode à un pas puisque seul le pas de temps précédent est pris en compte dans la résolution.

La position u et la vitesse v du point au temps $t + dt$ s'expriment de la manière suivante :

$$v_{t+dt} = v_t + \gamma_t dt$$

$$u_{t+dt} = u_t + v_t dt$$

Cette méthode d'intégration est donc très rapide à calculer, mais converge assez mal. Les méthodes de Runge Kutta dont l'ordre peut varier de 1 à 4 sont généralement plus stables et plus précises, à partir de l'ordre 2. Le meilleur compris est la méthode de Runge Kutta à l'ordre 4 qui offre un maximum de stabilité pour un coût de calcul raisonnable. Toutefois, elle n'est toujours pas adaptée dans le cas de problèmes *raides*, comme la simulation de vêtement à l'aide d'un système masses-ressorts.

Ensuite, on retrouve plusieurs méthodes d'intégration à pas multiples (utilisation des valeurs connues à n pas antérieurs) qui améliorent la qualité des valeurs obtenues mais au détriment de la stabilité.

Les méthodes implicites

Les méthodes implicites, quant à elles, utilisent des informations qui ne sont pas encore disponibles. Ainsi, pour calculer une vitesse à l'instant $t + dt$, on utilise l'accélération que le point aurait à $t + dt$ si les forces appliquées restaient les mêmes. Ces méthodes sont généralement légèrement plus lentes. En effet, les équations à résoudre sont souvent non linéaires, la résolution est donc plus difficile. Une solution est de linéariser les équations, comme le font Baraff et Witkin [BW98] pour l'animation de vêtements. Quelle que soit la solution retenue, la perte de temps occasionnée par les calculs plus lourds est largement compensée par une augmentation de la stabilité de la résolution et par l'utilisation d'un pas d'intégration plus grand, même dans le cas de problèmes raides comme la simulation de textiles.

Les méthodes hybrides

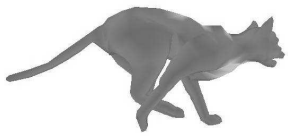
Enfin, des méthodes hybrides existent. Il s'agit dans ce cas d'utiliser les valeurs en cours de calcul pour *corriger* l'intégration. Cela augmente la stabilité de la résolution. Par exemple, en faisant dépendre la nouvelle position de la nouvelle vitesse que l'on calcule alors en premier, on trouve le schéma d'Euler modifié :

$$v_{t+dt} = v_t + \gamma_t dt$$

$$u_{t+dt} = u_t + v_{t+dt}dt$$

Finalement, toutes les méthodes ont leurs avantages et leurs inconvénients, que ce soit au niveau des temps de calcul, de la stabilité de la méthode ou bien de la qualité des résultats obtenus. Il s'agit donc de choisir la méthode qui convient le mieux au problème spécifique que l'on cherche à résoudre.

En bref : Les modèles générateurs permettent de simplifier les animations lorsque le nombre d'objets à animer est important car, au lieu de décrire en détail les différentes formes et positions de l'objet au cours du temps, il suffit de spécifier les équations le faisant évoluer et de les résoudre. Par contre, ces modèles sont beaucoup moins intuitifs pour un utilisateur non expérimenté, et souvent, les paramètres physiques du modèle (raideur, masse) sont difficiles à appréhender. Il est donc généralement nécessaire de fournir un environnement de développement adapté et simple pour l'animateur pour lui permettre de construire le modèle et de choisir des paramètres mécaniques réalistes et adaptés. De nombreux infographistes préfèrent rester maîtres de l'animation dans ses moindres détails et beaucoup préféreront des méthodes géométriques qui laissent libre cours à la créativité. Lorsqu'il s'agit de modélisation de personnages pour l'animation, il faut aussi garder à l'esprit que les personnages mis en scène sont très souvent non réalistes et ont des comportement exagérés, inobservables dans la réalité.
Quelle que soit la technique de déformation et le type de surface utilisés, l'état courant du modèle doit être affiché à l'écran et donc, converti en polygones. Cette conversion n'est pas toujours immédiate. En effet, lorsque l'on travaille avec des surfaces implicites par exemple, la surface nécessite d'être triangulée avant d'être affichée (marching cubes, marching triangles, technique des graines), ce qui peut prendre dans certains cas beaucoup de temps. Il est donc important de choisir une bonne modélisation de l'objet suivant les contraintes que l'on souhaite satisfaire (temps-réel par exemple) et une technique de déformation adaptée.



CHAPITRE 3

Modélisation et déformation des muscles, tissus et peau de personnages

Ce chapitre se propose de faire un état de l'art des modèles ayant été utilisés pour modéliser des muscles, des tissus ou de la peau de personnages. Nous commencerons par une section sur l'anatomie qui nous servira à mieux comprendre les modèles anatomiques, puis nous parcourrons tout d'abord les modèles surfaciques qui se contentent de déformer uniquement une *peau*. Ensuite, nous étudierons les modèles en couches et en particulier, les différentes techniques de modélisation de muscles et de tissus volumiques, ainsi que des surfaces utilisées pour recouvrir ces structures. Enfin, nous nous intéresserons aux détails externes qui peuvent être rajoutés aux différents modèles afin d'en améliorer le réalisme, comme par exemple, les plis de la peau ou de vêtements.

3.1 Le point de vue anatomique

Certains modèles de personnages de synthèse (humanoïdes, animaux) s'inspirent de l'anatomie pour rester plus proches de la réalité et ainsi obtenir des résultats plus réalistes. Cette introduction sur l'anatomie va nous permettre de mieux comprendre certains modèles anatomiques et physiques et de définir quelques termes utiles pour la description de ces modèles.

3.1.1 Définitions

Muscles

Sont qualifiés de *squelettiques* [SPCM97, Ned98, NT98, Mau99, NT00, Aub02] les muscles servant à faire bouger les os. Parmi ces muscles, les plus simples et les plus nombreux sont les muscles *fusiformes* qui sont composés d'une partie contractile, le *ventre* ou *chef* du muscle, et de deux extrémités,

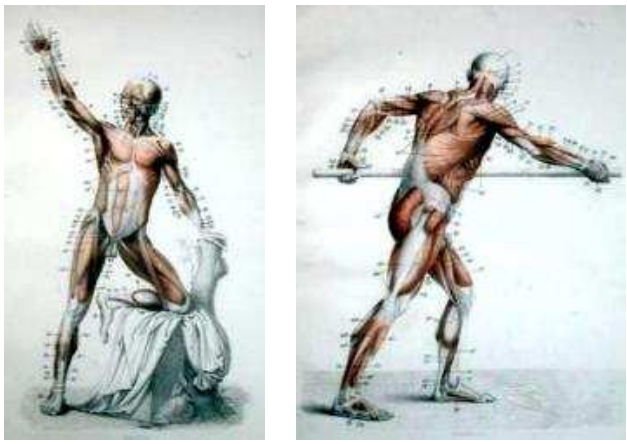


FIG. 3.1 – Musculature réelle.

les *tendons*, qui sont les *insertions* du muscle sur les os. Cependant, les deux insertions sont souvent différenciées. Celle qui est la plus proche de la colonne vertébrale est désignée par *insertion proximale* ou *origine* (généralement le point d'attache qui ne bouge pas) et celle qui est la plus éloignée sera appelée *insertion distale* ou tout simplement *insertion* (l'extrémité qui bouge le plus).

Par ailleurs, les anatomistes considèrent deux types de contraction :

- la contraction *isométrique* (même longueur) : dans ce cas, le muscle se contracte, le ventre change de forme mais ne change pas de longueur et ne produit pas de mouvement.
- la contraction *isotonique* (même tonicité) : dans ce cas, le ventre du muscle change de forme, la longueur du muscle change, ce qui induit un mouvement des os.

En général, les efforts musculaires font intervenir les deux types de contraction. Cependant, nous verrons dans la suite que le plus souvent, seule la contraction isotonique est présente dans les modèles d'animation car c'est la seule contraction qui induit un mouvement du squelette. C'est aussi la contraction qui se voit le plus, puisque le muscle rétrécit et gonfle.

Peau

La peau est composée de 3 couches schématisées sur la figure 3.2 : l'épiderme (1), le derme (2) et l'hypoderme (3).

L'épiderme recouvre le derme et est la couche superficielle de la peau. C'est cette couche qui comporte des irrégularités que l'on peut appeler *rides de petite échelle*.

Le derme, composé de collagène et d'élastine, est la couche principale de la peau. Le collagène lui confère ses propriétés de résistance à l'étirement, alors que l'élastine est responsable de l'élasticité de la peau.

Enfin, on trouve l'hypoderme en dessous du derme. Il est responsable de la jonction entre le derme et les couches sous-cutanées (muscles, os). Il se compose principalement de tissus graisseux et d'une couche contenant des vaisseaux sanguins et des nerfs qui permet à la peau de glisser sur les structures sous-cutanées.

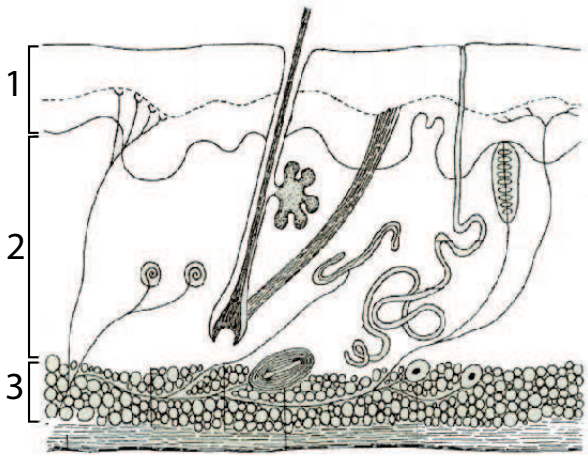


FIG. 3.2 – Composition de la peau : (1)-épiderme ; (2)-derme, (3)-hypoderme.

3.1.2 Propriétés mécaniques

Muscles

Le *ventre* du muscle est composé d'un ensemble de fibres parallèles, organisées en fuseaux, qui sont des composants élastiques capables de se contracter et de se relâcher. Dans [CHP89], on retrouve les deux propriétés mécaniques suivantes :

- Élasticité :
  L'élongation élastique est la capacité du muscle à revenir à sa position initiale après contraction. Mécaniquement, cela s'exprime de la manière suivante : lorsque l'on tire sur un muscle de longueur initiale L, de section A avec une force F, il en résulte une élongation E définie par l'équation :

$$E = (F * L * k)/A \tag{3.1}$$

  avec k un scalaire caractéristique du muscle.
- Contractibilité :
  C'est la capacité d'un muscle à raccourcir par stimuli nerveux. C'est un processus actif, contrairement à l'élongation élastique. Une relation constante existe entre la longueur naturelle du muscle, la longueur variable du muscle contracté et le degré de rotation à l'articulation. Soit R, l'unité de rétrécissement (i.e. longueur du muscle au repos - longueur courante), soit θ, l'angle de rotation de l'articulation et soit k le scalaire caractérisant le muscle :

$$R = \theta * k \tag{3.2}$$

Les tendons aussi sont flexibles et élastiques. Cependant, étant donné qu'ils le sont à un autre ordre de grandeur par rapport aux muscles, on fait souvent l'hypothèse que leur longueur ne change pas pendant le mouvement.

Peau

La peau est essentiellement élastique et quasi-incompressible. C'est pourquoi elle peut s'étirer, mais plisse lorsqu'elle se retrouve compressée afin de garder une aire constante. Elle est aussi plastique puisqu'au cours du temps, elle subit aussi des déformations irréversibles (rides de vieillesse).

3.2 Habillage géométrique surfacique et ses déformations

Cette section décrit les techniques utilisées pour animer des modèles surfaciques. Il en existe trois principales pouvant être appliquées sur des maillages polygonaux : l'interpolation de formes clefs (décrite à la section 2.3.1.1), le *skinning* (décrit à la section 2.3.1.2) et les déformations de forme libre (décrites à la section 2.3.1.3). Ces techniques étant les plus utilisées dans l'industrie (jeux vidéos ou films), de nombreux travaux de recherche portent sur l'amélioration et l'automatisation de l'une ou de l'autre de ces approches.

3.2.1 Interpolation de formes clefs

Pose Space Deformation

Pose Space Deformation est une technique de déformation de maillage introduite par Lewis et al. [LCF00] qui généralise l'interpolation de formes clefs. Cette approche se place totalement du côté de l'infographiste qui doit sculpter son personnage dans différentes poses, les formes clefs. Ces formes clefs possèdent différents contrôles qui peuvent être de nature très variée : angle d'une articulation d'un squelette, expression d'un visage (joie, tristesse), âge du personnage. Le nombre de contrôles pour chaque point du maillage n'est pas limité. Les contrôles des formes clefs servent à définir un espace de paramètres où chaque contrôle représente un axe et où chaque pose peut être positionnée sur chaque axe.

Pendant l'animation, la surface déformée est obtenue par interpolation des points du maillage à travers les différents contrôles. La surface correspondant à une pose donnée est définie par $p + \vec{\delta}$, p étant la position du point entraîné rigidement par le squelette et $\vec{\delta}$, la différence entre la position obtenue et la position de la pose associée. L'algorithme fonctionne de la manière suivante :

1. L'artiste positionne des contrôles et dessine une enveloppe.
2. Pour chaque pose, les points qui ont bougé par rapport à la position de repos sont recherchés et on calcule le $\vec{\delta}$ correspondant pour chaque point. Par définition, $\vec{\delta} = 0$ dans le cas d'une transformation rigide.
3. Pour chaque sommet du maillage, on a donc un ou plusieurs $\vec{\delta}$.
4. Lors de l'animation, on doit déterminer la position de chaque sommet pour la pose courante. Elle est égale à la concaténation des degrés de liberté relatifs des contrôles interprétés comme des dimensions indépendantes. Chaque $\vec{\delta}$ est interpolé par une fonction à base radiale, calculée comme une combinaison linéaire de fonctions de distance non linéaires entre les données.

Shape by example

Sloan et al. [SIC01] reprennent la même technique, mais n'effectuent pas l'interpolation de la même façon. En effet, au lieu de résoudre un système linéaire par degré de liberté (par point), ils proposent de résoudre un système linéaire par pose, ce qui s'avère être moins coûteux. Une fois

l'espace paramétrique défini, ainsi que les poses, ils associent une fonction de base à chaque exemple de pose fourni par l'utilisateur, en s'assurant que cette fonction ait une valeur de 1 pour l'exemple en question, et de 0 pour tous les autres exemples. Pendant l'animation, la pose courante est donnée par l'addition des fonctions de base (une par position clef). Celles choisies par Sloan et al. sont des combinaisons de fonctions à base radiale et de fonctions linéaires de la forme :

$$w(p) = \sum_{i=0}^{N} r_i R_i(p) + \sum_{j=0}^{D} l_j L_j(p)$$

avec :
N, le nombre d'exemples ;
D, le nombre de contrôles ;
r_i et R_i, les poids et les fonctions radiales ;
l_j et L_j, les poids et les fonctions linéaires.

Ainsi, pour chaque position p dans l'espace des paramètres, on peut trouver la valeur de chaque contrôle :

$$x(p) = \sum_{i=0}^{N} w_i(p) x_i$$

Les deux approches présentées ci-dessus utilisent des poses créées par des artistes comme poses clefs. le résultat obtenu étant fidèle aux exemples clefs, il dépend surtout des talents de l'animateur qui a créé ces poses. De plus, les maillages clefs créés doivent être de même topologie et contenir le même nombre de sommets si l'on veut que l'interpolation soit possible. On peut aussi noter qu'en contrepartie de la liberté de modélisation laissée à l'animateur, l'effort et le temps de modélisation des différentes poses peut s'avérer décourageant.

Deformation from range scan data

C'est pourquoi Allen et al. [ACP02] utilisent comme poses clefs des données capturées par un scanner $3D$. Le défi, en dehors des problèmes liés à la reconstruction de la surface et l'extraction d'un squelette, provient du fait qu'il n'est pas possible d'interpoler directement les données reconstruites, celles-ci étant simplement un ensemble de points, d'un nombre variable selon les poses, et comportant un certain nombre de trous. Les auteurs utilisent donc une surface de subdivision *modèle* qui permet de capturer, pour chaque pose, une carte de déplacements qui traduit les différences entre le *modèle* et la pose réelle capturée. Cette représentation appelée *displaced subdivision surface* a été introduite par Lee et al. [LMH00]. Ensuite, les trous dans les poses capturées sont bouchés au niveau des cartes de déplacements par des méthodes d'optimisation. Pendant l'animation, les déplacements à appliquer au modèle pour déterminer la pose courante sont interpolés à l'aide des déplacements des k plus proches voisins parmi les points des exemples clefs. Pour chaque point du *modèle*, il se calcule par une combinaison linéaire des points des k poses sélectionnées, en utilisant des poids normalisés variant de 0 à 1.

Plus récemment, Kurihara et al. [KM04] utilisent des données médicales pour reconstruire et animer une main. Afin de former le maillage et d'obtenir les poses clefs, ils utilisent un maillage de référence qu'ils déforment ensuite successivement par *skinning*, par une déformation à l'aide d'une fonction à base radiale puis par une mise en correspondance entre certains points sélectionnés entre le maillage de la pose clef et le maillage scanné. L'interpolation entre les formes clefs se fait ensuite en utilisant une *pose space deformation* comme introduite par [SIC01], mais en y rajoutant un certain

nombre de poids pour tenir compte du fait que tous les points dans une pose ne sont pas influencés de la même façon par tous les éléments du squelette.

Les techniques basées sur l'interpolation de formes clefs permettent de générer des mouvements cinématiques (i.e. composés d'une succession d'états d'équilibre) généralement très réalistes puisqu'elles s'inspirent de poses où la déformation du maillage est correcte. Il n'est néanmoins pas possible de générer des effets dynamiques. Quels que soient la vitesse et le type de mouvement qui ont conduit à une pose donnée, la forme générée sera la même. Un avantage non négligeable est que les poses courantes peuvent être synthétisées en temps-réel. Cependant, le gros inconvénient des interpolations de formes clefs est l'espace de stockage requis pour stocker les poses elles-mêmes. C'est pourquoi des techniques plus orientées vers le *skinning* essaient de garder le principe d'utiliser des formes clefs pour en extraire des repères ou des poids lors de pré-calculs, les poses n'étant plus utilisées au cours de l'animation.

3.2.2 Déformations dans les sous-espaces du squelette

Medial-based vertex deformation

Pour remédier aux problèmes classiques du *skinning* que nous avons mentionnés à la section 2.3.1.2, Bloomenthal [Blo02] utilise l'axe médian de l'objet pour calculer des repères de *skinning* intermédiaires et les poids associés. Il constate que le problème de rétrécissement d'une articulation (voir figure 2.16 (b)) lors d'un mouvement de rotation peut être diminué, voire évité si l'on rajoute suffisamment de repères de *skinning* entre les repères déjà existants, pourvu que les poids associés aux repères soient correctement calculés.

Dans l'approche de Bloomenthal, l'axe médian[1] de l'objet est représenté par un ensemble de triangles et de segments. Ce sont les *primitives* P_i. Un champ de convolution est produit autour de l'axe médian en convoluant un filtre cubique $3D$ avec cet axe. À chaque primitive P_i, on associe un repère de *skinning* F_i (par exemple, pour un triangle, il pourra correspondre au repère formé par la normale et un des côtés du triangle). Un point du maillage dépend de ce repère si la valeur de convolution est non nulle en ce point. Son poids w_i est alors la largeur du filtre de convolution (qui dépend du rayon de l'axe médian en ce point).

Une fois cette étape de pré-calculs effectuée (calcul de l'axe médian pour trouver les repères de *skinning* et calcul de la convolution pour obtenir les poids), il ne reste plus qu'à animer le modèle. L'animation se fait classiquement à partir d'un squelette interne. Par simple *skinning*, on recalcule la position des primitives de l'axe médian en fonction de l'influence des éléments du squelette (au lieu de calculer directement la position finale des sommets), ce qui donne une nouvelle position des repères intermédiaires F_i. Ensuite, on effectue une combinaison linéaire pondérée pour calculer la position des sommets du maillage à partir des F_i et w_i correspondants.

Cette technique simple permet, grâce à un pré-calcul lourd, de rajouter un niveau d'abstraction qui crée des repères de *skinning* intermédiaires. Les résultats sont de très bonne qualité pour les zones à problèmes. Le point négatif de cette méthode se situe au niveau du calcul de l'axe médian, qui, bien qu'il existe toujours et de façon unique, peut parfois être long et difficile à calculer.

Skins from examples

Mohr et Gleicher [MG03] proposent une technique pour améliorer le *skinning* en ajoutant, pour les mêmes raisons que Bloomenthal [Blo02], de nouveaux repères au squelette. Cependant, cette fois,

[1] L'axe médian d'un objet est le lieu des centres des sphères maximales incluses dans l'objet.

ces repères peuvent être tournés, translatés ou bien agrandis pendant l'animation, ce qui, en plus d'éviter les problèmes classiques liés au *skinning*, permet d'ajouter des effets tels que le gonflement des muscles. Les nouveaux repères et leurs poids associés (un poids par sommet par repère qui influence ce sommet) sont déterminés à partir d'un ensemble d'exemples dans des poses extrêmes, par résolution d'un système aux moindres carrés. L'avantage de cette technique basée sur l'observation d'exemples, par rapport aux techniques semblables qui utilisent l'interpolation de formes, est qu'ici, les formes clefs ne sont pas stockées. Elles servent seulement à un pré-calcul de poids. Ensuite, un *skinning* classique est appliqué au cours de l'animation, en prenant en compte les repères supplémentaires. L'inconvénient majeur de cette technique se situe au niveau de l'ajout des nouveaux repères. Bien que se basant sur des exemples clefs, leur technique ne donne pas toujours des résultats satisfaisants et l'animateur doit, dans la plupart des cas, les retravailler, voire même, les placer entièrement.

Multi-weight enveloping

À la différence des deux articles précédents, Wang et al. [WP02] ne cherchent pas à ajouter des nouveaux repères, mais des nouveaux poids. Ainsi, au lieu d'avoir un poids par sommet par repère, ils utilisent 12 poids par point par repère, un pour chaque coefficient de la matrice de transformation du repère. Cela augmente légèrement le nombre de données à stocker et crée un surcoût au niveau de la combinaison linéaire au moment de l'animation. Cependant, avoir un poids par coefficient de la matrice de transformation permet de prendre en compte une transformation supplémentaire de la matrice (rotation, translation, agrandissement), différente pour chaque point et ainsi de capturer des détails supplémentaires. Les poids sont déterminés à partir de poses clefs, générées à la main ou bien par simulation physique, par une minimisation aux moindres carrés.

EigenSkin

Kry et al. [KJP02] présentent une technique qui permet d'obtenir des déformations non linéaires, quasi-statiques, de personnages articulés. Ceci est fait en composant un champ de déplacements calculé en temps-réel (*EigenSkin*) à un *skinning* classique, l'idée étant de déformer de façon réaliste et rapide le maillage de l'objet avant d'appliquer le *skinning*. Pour calculer le champ de déplacements, l'objet doit être simulé de façon réaliste dans au moins 6 poses différentes pour chaque degré de liberté. Pour cela, les auteurs utilisent un modèle d'éléments finis de la main avec un squelette interne. Ils pourraient aussi bien utiliser un autre modèle de simulation réaliste coûteux. Cela permet d'obtenir des positions ν_i^p des sommets observés dans la pose p_i.

Soit $\mathbb{P}$, l'ensemble des indices p des poses observées. p_0 représente la position de repos et ν_i^0, la position du sommet i dans la position de repos. ν_i^p représente la position du sommet observé dans la pose p_i. Le déplacement u_i^p, i.e. la différence entre la position du sommet que l'on aurait obtenue en appliquant un *skinning* classique et la position de la pose observée, s'exprime dans la position de repos par :

$$u_i^p = (\sum_{b \in B_i} w_{ib} T_b^p)^{-1} \nu_i^p - \nu_i^0$$

Concrètement, cela revient à appliquer la transformation inverse du *skinning* au point observé, ce qui nous donne un objet déformé pour lequel on peut calculer, pour chaque sommet, une différence par rapport à la position de repos. En stockant ces déplacements par rapport à la position de repos, il est possible de modifier le maillage avant d'appliquer le *skinning* pour retrouver la position correcte de chaque sommet.

Ces déplacements sont donc calculés pour chaque sommet, pour chaque degré de liberté de l'articulation qui l'influence. Ceci est déterminé en faisant bouger chaque articulation indépendamment, et en cherchant les points influencés (il ne sont évidemment pas les mêmes que l'ensemble des points attachés à l'articulation et ces ensembles peuvent se recouvrir). Cet ensemble de points est appelé le *support* de l'articulation. Pour chaque point du support, on fait une analyse en composantes principales de ses déplacements. La méthode permet aussi bien de corriger la position des sommets du maillage que les normales utilisées pour le rendu.

Une fois ces pré-calculs effectués, le modèle peut-être animé en temps réel. Il suffit, pour chaque nouvelle pose, de calculer la correction à appliquer au maillage (il est possible d'utiliser de 1 à 10 valeurs propres, 5 si l'on applique aussi une correction des normales), suivie d'un *skinning* classique. Cette technique a été implémentée sur une carte graphique (en utilisant l'extension *vertex program*) et a permis de réaliser ces déformations en temps-réel.

3.2.3 Cas des modèles paramétriques

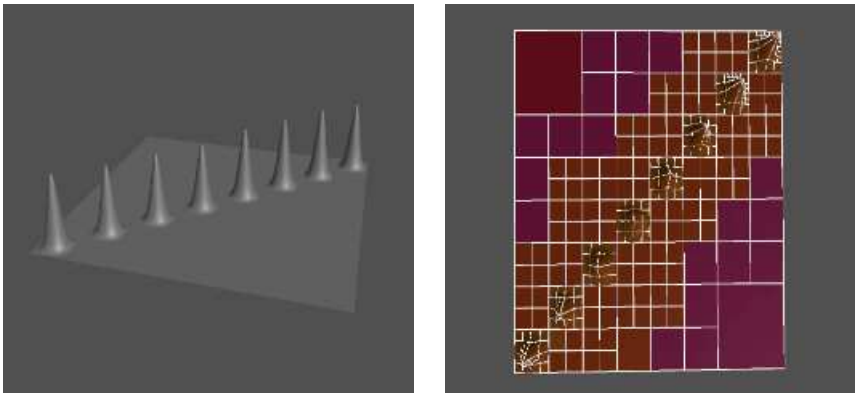
Très agréables pour modéliser, les surfaces B-splines posent néanmoins des problèmes lorsqu'il s'agit d'animer des modèles très détaillés. Le principal inconvénient lorsque l'on veut rajouter des détails au modèle, est qu'il faut rajouter des points de contrôle là où l'on veut des détails. Mais, pour des raisons de continuité, il faut alors obligatoirement découper l'ensemble de la ligne de points selon chaque domaine paramétrique. Concrètement, cela se traduit par le découpage d'une ligne entière de carreaux ou d'une colonne entière. De ce fait, de nombreux points de contrôle sont rajoutés et ils sont le plus souvent inutiles en dehors du carreau ou l'on souhaitait mettre des détails. Le modèle se retrouve donc trop détaillé à certains endroits ce qui rend très difficile l'animation de déformations à grande échelle.

B-splines hiérarchiques

L'idée de Forsey introduite dans [FB88] et étendue à l'animation dans [For91], est de ne subdiviser que les carreaux de B-splines où cela est nécessaire, sans subdiviser les carreaux des lignes et colonnes adjacentes (voir figure 2.4). Pour cela, lorsqu'un carreau est subdivisé, il est défini localement comme un déplacement ou *overlay* par rapport au carreau non divisé sous-jacent. L'exemple figure 3.3 montre le gain qui est fait en prenant un exemple où il aurait fallu, avec des carreaux de B-splines classiques, diviser entièrement la surface en carreaux les plus fins afin de pouvoir modéliser les pics. Grâce aux B-splines hiérarchiques, on ne divise que les carreaux de la diagonale.

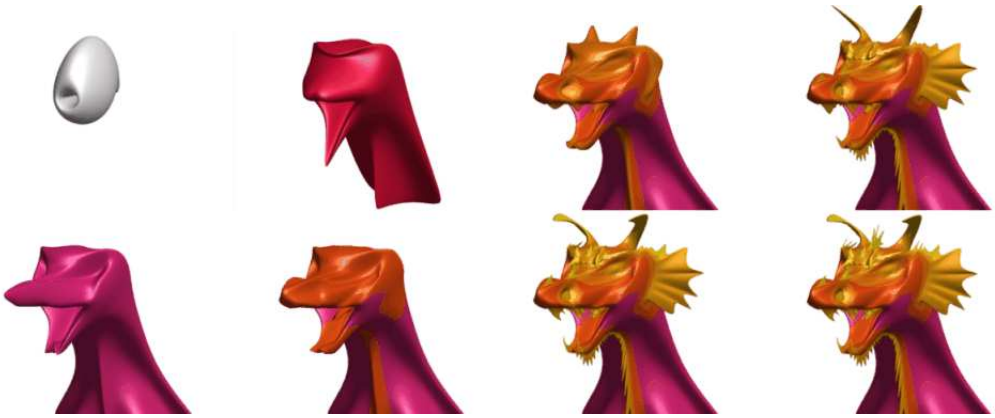
Une surface B-spline hiérarchique est créée par l'application successive d'un raffinement local d'une surface B-spline initiale (un plan, un cylindre ou un tore). Par exemple, la tête du dragon figure 3.4 a été construite à partir d'un tore. La figure montre les 8 niveaux de la hiérarchie, en commençant par le tore. On obtient finalement un modèle très détaillé à un faible coût.

Pour l'animation, le modèle est constitué d'un squelette interne et de la peau qui est une surface B-spline hiérarchique. L'utilisateur se charge de relier les points de contrôle, à n'importe quel niveau, aux différents os du squelette qui doivent l'influencer. Ensuite, le mouvement du squelette entraîne, de manière rigide, les points de contrôle. Le dragon figure 3.5 est animé en bougeant un squelette interne auquel est attachée la *peau*. L'avantage est que les détails suivent le changement de forme de la surface à grande échelle. Si les détails ne changent pas, ils n'ont pas besoin d'être animés, ils restent identiques et suivent le niveau le plus grossier qui bouge.



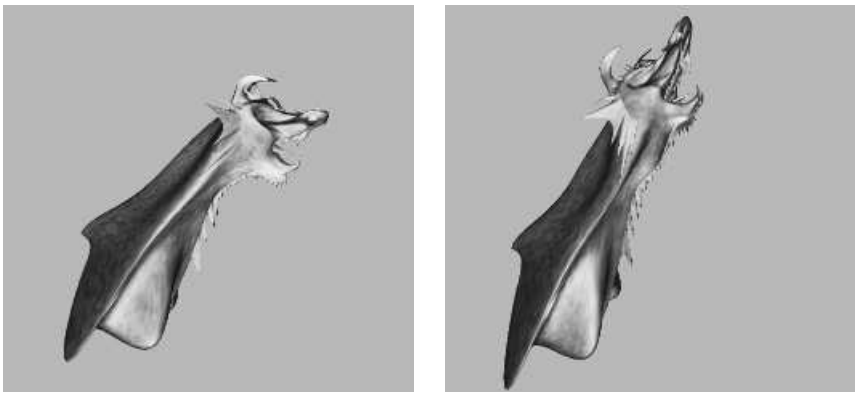
(a) (b)

FIG. 3.3 – Mise en valeur du gain des B-splines hiérarchiques par rapport aux B-splines classiques : (a)-surface plate avec une série de bosses sur la diagonale ; (b)-modèle présenté en (a) construit avec des B-splines hiérarchiques.



(a) (b)

FIG. 3.4 – Un modèle de tête de dragon et ses niveaux de raffinement : (a)-les quatre premiers niveaux (le modèle de départ étant un simple tore) ; (b)-les quatre derniers niveaux avec un modèle de plus en plus détaillé.



(a) (b)

FIG. 3.5 – Un exemple d'animation utilisant les B-splines hiérarchiques : (a)-modèle au repos ; (b)-modèle dans la pose courante de l'animation.

Surface Oriented Free Form Deformations

Singh [SK00] propose des FFDs surfaciques (voir section 2.3.1.3 pour une description des FFDs classiques). Ainsi, le volume de contrôle des FFDs classiques est remplacé par une surface constituée de gros polygones et recouvrant la surface polygonale (ou paramétrique) du modèle. L'espace déformé par la surface de contrôle est alors défini par une fonction de distance autour de cette dernière.

L'algorithme se déroule en trois phases. La première consiste à placer la surface de contrôle sur la surface déformable puis à *figer* les deux surfaces dans une position de repos. Ensuite, la phase d'*enregistrement* consiste à calculer, pour chaque sommet du maillage déformable un ensemble de poids qui décrivent comment chaque polygone de la surface de contrôle influence chaque sommet. Le poids est inversement proportionnel à la distance du point P au polygone de contrôle concerné et ce poids est réajusté à l'aide d'un coefficient de localité qui permet de contrôler la zone d'influence de la déformation autour du point qui est déplacé. En plus de ces poids qu'il faut normaliser, la position de chaque sommet P est localement calculée dans chaque polygone de contrôle .

Une fois ces pré-calculs effectués, l'utilisateur peut modifier la surface de contrôle et ainsi indirectement modifier la surface déformable. C'est la phase de *déformation*. En effet, chaque point P a une nouvelle position dans les polygones de contrôle et la position retenue est simplement la combinaison linéaire pondérée de ces positions. Cette technique se déroule de la même façon qu'un *skinning* (section 2.3.1.2), les repères des articulations étant remplacés par les repères des polygones de contrôle.

L'avantage de ces FFDs par rapport aux FFDs classiques est que même si la manipulation de la surface déformable reste indirecte, la surface de contrôle manipulée ressemble beaucoup à la géométrie qu'elle déforme. Cela rend donc l'outil intuitif pour l'utilisateur, tout en ayant des propriétés de conservation de continuité de la surface. Cependant, il s'avère que la position du maillage de contrôle et la taille de ses éléments jouent un rôle important en ce qui concerne la qualité des résultats.

Cette technique à été appliquée avec succès pour déformer les régions de la jambe et du bassin d'un personnage, zones qui posent généralement des problèmes avec l'utilisation d'un *skinning* par exemple. Des résultats sont montrés sur la figure 3.6.

FIG. 3.6 – Deux exemples d'utilisation des FFDs surfaciques. Sur l'image de gauche, la chemise du personnage est déformée quand son bras bouge. Sur l'image de droite, on peut voir une déformation correcte de la zone du bassin du personnage et de sa jambe (images extraites de [SK00]).

En bref : Parmi les outils qui permettent de simplifier le travail de l'animateur, on trouve tout d'abord l'interpolation de forme clefs. Grandement explorée, cette technique souffre néanmoins toujours des mêmes problèmes. En effet, afin d'obtenir de bons résultats, l'artiste doit créer les bonnes poses, en parcourant tous les degrés de liberté possibles de la déformation, ce qui n'est pas toujours évident. De plus, déformer directement une surface à la main s'avère être une tâche difficile et entraîne souvent des discontinuités de la surface déformée. D'autre part, ces formes clefs doivent être stockées pendant l'animation ce qui rend le procédé très coûteux en place mémoire, même si les animations peuvent être créées en temps-réel. Lorsque que les poses clefs sont issues de données scannées ou médicales, il faut en plus faire face à des problèmes topologiques et de bruit.
Les FFDs sont également très utilisées puisqu'au lieu de déformer la surface directement, l'utilisateur manipule un objet de plus haut niveau, comportant moins de points de contrôle. Cette technique reste bien adaptée lorsqu'il s'agit de créer quelques déformations du modèle (des formes clefs par exemple), mais se retrouve impraticable pour créer une animation complète.
En effet, l'utilisation d'un squelette interne habillé par *skinning* facilite la manipulation des surfaces déformables pour créer des déformations réalistes. Le nombre de repères à manipuler est bien inférieur au nombre de sommets du maillage, ce qui rend cette technique très populaire, malgré les défauts classiques de la technique. De plus, de nombreux papiers proposent des solutions en ajoutant des repères, des poids ou en se basant sur des acquisitions réelles. Une fois les ensembles repères/poids calculés, le coût de stockage est faible et l'animation est très rapide. C'est pourquoi cette méthode est très intéressante dans notre cas puisqu'elle apporte une solution peu coûteuse pour calculer les déformations cinématiques de la peau d'un personnage.
Actuellement, le seul inconvénient de cette technique est que, comme l'interpolation de formes clefs ou bien les FFDs sans masses-ressorts, elle ne permet pas de créer une réponse dynamique de la surface aux mouvements.

Avant que ces solutions récentes d'amélioration du *skinning* ne soient proposées, des modèles en couches plus complexes ont été élaborés pour pallier au manque de réalisme du *skinning*. Le principe de ces modèles est de s'inspirer de l'anatomie en modélisant, une par une, les couches constituant un

être vivant. Ainsi, un modèle $3D$ n'est plus simplement une surface géométrique animée par son squelette, mais il se compose, en plus du squelette, d'une couche musculaire où les muscles sont modélisés un à un, éventuellement d'une couche de tissus adipeux, et d'une couche de peau élastique pour recouvrir l'ensemble. À ces trois couches, qui sont présentes individuellement ou fusionnées, il est possible d'ajouter une quatrième couche modélisant les cheveux ou les vêtements. Par ailleurs, nous souhaitons aussi mentionner que dans certains modèles, le squelette est modélisé par des os (représentés par des maillages triangulaires par exemple). Ainsi, plus le modèle est détaillé, plus les résultats obtenus sont corrects. Cependant, la qualité des résultats se paie par des étapes de modélisation, puis d'animation des différentes couches, plus complexes et plus longues à réaliser.

3.3 Modélisation des muscles pour un modèle en couches

Nous présentons dans cette section différents modèles de muscles ayant été utilisés pour simuler la cinématique[2] ou la dynamique de muscles individuels. Nous avons regroupé ces différents modèles en deux catégories, les modèles géométriques et les modèles physiques, car le type de modèle de déformation utilisé est directement lié au genre de mouvements qu'il est possible de simuler.

3.3.1 Modèles géométriques

Les modèles géométriques, comme leur nom l'indique, ne se basent que sur des données géométriques de l'objet connues à un instant donné et par conséquent, ne peuvent que calculer des déformations cinématiques (pas d'effet dynamique). Les modèles que nous allons présenter sont généralement greffés sur un modèle en couches dont le squelette est la couche inférieure et la peau la couche supérieure. Le principe de l'algorithme reste ensuite à peu près toujours le même : les *os* bougent et entraînent un gonflement des muscles qui permet d'obtenir une déformation réaliste de la peau. Les différences d'un modèle à l'autre se retrouvent au niveau du type de surface et de la technique de déformation utilisées.

Paire de Free Form Deformations

En 1989, Chadwick et al. [CHP89] introduisent pour la première fois la notion de modèle en couches. Le modèle est simple et ne se compose que de deux couches : une couche contenant plusieurs muscles et la peau, fusionnée avec la couche externe des muscles réunis.

Représentation d'un muscle

Un muscle individuel est représenté par un maillage polygonal enfermé dans une paire de FFDs (voir section 2.3.1.3), comme on peut le voir sur la figure 3.7.

La concaténation des deux volumes paramétriques donne 7 plans perpendiculaires à l'axe de l'articulation (4 plans pour chaque FFD dont un plan commun pour les deux). Les deux plans situés à chaque extrémité sont les *adjoining planes* et servent à préserver la continuité entre deux morceaux de muscles adjacents. Les trois plans du milieu, appelés *mid planes* par les auteurs, servent à modéliser le comportement du muscle, en réponse à la cinématique du squelette. Ce modèle de muscle est utilisé pour les déformations statiques des bras et des jambes (tendons + muscles) d'un humanoïde. Le principe de l'algorithme est le suivant : le squelette bouge et crée automatiquement la flexion

[2]Par *cinématique*, nous entendons toutes les déformations statiques successives du modèle, sans phénomène vibratoire.

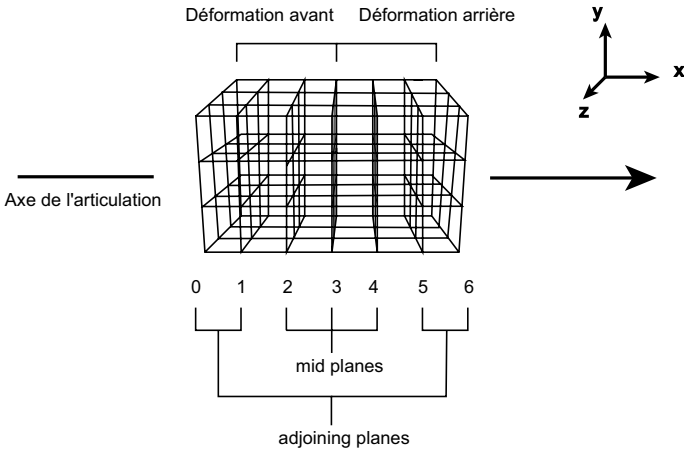


FIG. 3.7 – Muscle proposé par Chadwick et al., modélisé à l'aide d'une paire de FFDs (image inspirée de [CHP89]).

appropriée des muscles. Nous remarquerons que si ce modèle est adapté pour modéliser des muscles fusiformes, il l'est beaucoup moins pour des muscles plus plats ou en triangle tels ceux du dos.

Déformation lors de l'animation

Les déformations statiques du muscle sont calculées à partir des propriétés physiques réelles d'élasticité et de contractibilité (voir équations (3.1) et (3.2) de la section 3.1). Le premier modèle de Chadwick et al. [CHP89] permet de modéliser les flexions/extensions d'un muscle, c'est à dire un muscle simple qui rétrécit ou s'agrandit suivant le rapprochement ou l'éloignement de ses extrémités.

L'algorithme se déroule de la façon suivante :

1. Calcul du rétrécissement du muscle à partir de l'angle de l'articulation.
2. Propagation de ce rétrécissement à travers chaque plan de contrôle du muscle. Pour chaque *mid plane*, le rétrécissement imposé sur x est compensé par un agrandissement du muscle le long des axes y et z (se reporter à la figure 3.7). Dans le cas général, cet agrandissement est le même dans toutes les directions pour chacun des trois *mid planes*. Mais si l'un des côtés est fixe, l'allongement ne pouvant se faire dans cette direction, on double l'allongement qui se produit dans la direction opposée. C'est ce qui se passe quand un muscle est collé à un os. Le rapport des rayons détermine les facteurs d'échelle des deux *mid planes* extérieurs (2 et 4). Le plan central (3) est déformé de telle sorte que la continuité soit maintenue sur toute la surface contractée.
3. Les *adjoining planes* sont légèrement déformés pour maintenir la continuité avec les autres muscles actifs, ou bien, ils restent non déformés en dehors du raccourcissement sur x.

Pour modéliser les muscles ou les morceaux de peau qui recouvrent une articulation, Chadwick et al. utilisent l'algorithme de déformation des tendons. Cette fois, il s'agit d'un phénomène passif d'élongation qui se retrouve sur de courtes régions couvrant une articulation (le dessous du coude par exemple).

1. Détermination de l'angle de bissection de l'articulation (avec utilisation d'un seuil), i.e., $angle = angle/2$. Pour les articulations à plusieurs degrés de liberté, il faut connaître l'axe de rotation résultant (utilisation des quaternions).

2. Les *mid planes* 2 et 3 sont tournés de l'angle de bissection autour de leur axe z et le plan 4 par l'opposé.
3. Les problèmes d'intersection sont évités en bougeant des points de contrôle si c'est nécessaire.
4. Éloigner les plans 0 et 1 de 2 et les plans 5 et 6 de 4.
5. Faire une rotation de l'ensemble des plans 4, 5 et 6.
6. Maintenir une continuité C^1 entre les différents points de contrôle tout en évitant les intersections entre les différents plans.
7. Utiliser le poids des points de contrôle pour maintenir le volume du muscle ou des tissus.

Un point positif de ce modèle est tout d'abord la continuité entre les muscles qui est facilement assurée grâce aux *adjoining planes* qui ne se déforment pas ou peu. Par contre, on peut noter un inconvénient au niveau de la peau. En effet, elle n'est pas flexiblement attachée aux structures du dessous mais elle est fusionnée avec elles, ce qui fait qu'elle subit les mêmes déformations que le muscle alors qu'elle devrait pouvoir glisser sur les muscles.

Surfaces implicites : utilisation d'ellipsoïdes

En 1997, deux papiers [SPCM97, Wil97] ont proposé des modèles de muscles identiques créés à partir d'ellipsoïdes. Les modèles de muscles présentés dans le papier de Scheepers et al. [SPCM97] se greffent sur un squelette constitué d'os rigides individuels construits par des primitives géométriques et les articulations sont modélisées par des fonctions. Dans le papier de Wilhelms [Wil97], les os et les tissus autres que les muscles sont représentés par des ellipsoïdes qui ne sont pas déformés au cours du temps. Par exemple, un os pourra être constitué de trois ellipsoïdes : un étiré et un sphérique dans chaque bout. Dans les deux cas, il est possible d'imposer un mouvement aux articulations qui engendre un mouvement des os, ce qui provoque une modification de la forme des muscles qui y sont attachés.

Représentation

Pour représenter le muscle (sans les tendons), Scheepers et al. [SPCM97] et Wilhelms [Wil97] utilisent des ellipsoïdes. Un ellipsoïde est une surface implicite définie par une fonction f de la forme :

$$f(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} + cste.$$

L'intérêt des ellipsoïdes est la facilité à pouvoir préserver un volume constant du muscle dans le cas d'une contraction isotonique, en contrôlant le rapport hauteur sur largeur de l'ellipse.

Les trois modèles de muscles présentés dans la suite utilisent des ellipsoïdes, dont les points d'attache aux os sont de nature différente :

- **Fusiform muscle model** [SPCM97, Wil97]
  C'est le modèle le plus simple. Il est composé d'un seul ellipsoïde et est attaché aux os par ses deux extrémités (origine, insertion). Il est défini par son volume v, son rapport r, sa position et son orientation. Il peut être muni de tendons auquel cas, soit deux nouvelles extrémités sont définies [SPCM97] (origine et insertion du *ventre*), soit deux petits ellipsoïdes sont ajoutés à chaque bout [Wil97]. Dans le premier cas, la longueur du tendon ne varie pas pendant le mouvement.
- **Multi-belly muscle model** [SPCM97]
  Certains muscles sont composés de plusieurs muscles fusiformes. Au lieu de les construire *à la main* en mettant côte à côte des muscles fusiformes, on peut utiliser le *multi-belly muscle*

model qui fait la même chose mais de manière automatique, quel que soit le nombre de *ventres* à positionner. Le muscle est alors composé de n origines et de n insertions. Comme illustré sur la figure 3.8 (a), une courbe spline pour l'origine et une courbe spline pour l'insertion sont utilisées et échantillonnées pour obtenir les origines et insertions des différents *ventres*. Pour ce type de muscle, on parle donc de courbe origine et de courbe insertion. Ensuite pour définir l'orientation de chaque *ventre*, on se sert de l'origine o_j et de l'insertion i_j correspondante. La direction perpendiculaire est alors la normale au plan formé par i_j et par les origines du voisin de gauche o_{j-1} et du voisin de droite o_{j+1} s'ils existent, o_j sinon.

– **General muscle model** [SPCM97]
Les deux modèles précédents ne permettent pas de représenter des muscles pliés. En effet, il sont adaptés pour des muscles fusiformes et des muscles plats. Les auteurs ont donc introduit le *general muscle model*. Il s'agit de découper le muscle en trois zones : origin-section, mid-section et insertion-section (voir figure 3.8 (b)). Les différentes sections sont délimitées par des points de contrôle : o_v pour séparer origine et mid sections et i_v pour séparer insertion et mid sections. Le muscle est constitué d'un ensemble de carreaux de surface formés à l'aide d'ellipses construites le long d'une courbe de Bézier cubique et de deux *chapeaux* (hémisphères elliptiques) mis à chaque bout pour *fermer* le muscle, mais dont le volume n'est pas pris en compte dans les calculs car négligeable par rapport au reste du muscle. Soient o_1 et o_2, les deux origines du muscle, i_1 et i_2, les deux insertions, o_c et i_c, respectivement les milieux de o_1o_2 et i_1i_2, et o_v et i_v, pour définir l'orientation du muscle. La courbe de Bézier est définie par les points de contrôle o_c, o_v, i_v et i_c. En o_c, l'ellipse a pour axes principaux : $a_0 = o_c - o_1$, et $b_0 = \frac{(o_v - o_c)}{2}$. De même, en i_c, l'ellipse a pour axes $i_c - i_1$ et $\frac{(i_v - i_c)}{2}$. En o_v et i_v, les axes de l'ellipse sont calculés à partir de v et de r pour préserver le volume et le rapport hauteur largeur du muscle. Ainsi, si les extrémités se rapprochent, le muscle gonfle et inversement si elles s'éloignent.

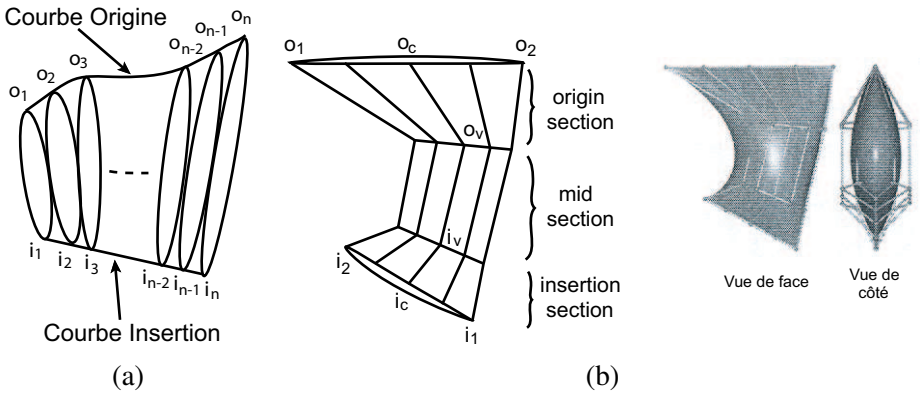


FIG. 3.8 – Modèles de muscles de Scheepers et al. : (a)-multi-belly muscle model ; (b)-general muscle model (images extraites de [SPCM97]).

Déformation du muscle

Les trois modèles de muscles de Scheepers et al. [SPCM97] sont capables de simuler les contractions isotoniques et les contractions isométriques alors que le modèle de Wilhelms [Wil97], plus limité, ne simule que les contractions isotoniques sur des muscles fusiformes (voir figure 3.9) :

– Cas d'une contraction isotonique [SPCM97, Wil97]
Lorsque les os bougent (se rapprochent ou s'éloignent), la longueur l du muscle change. Soit

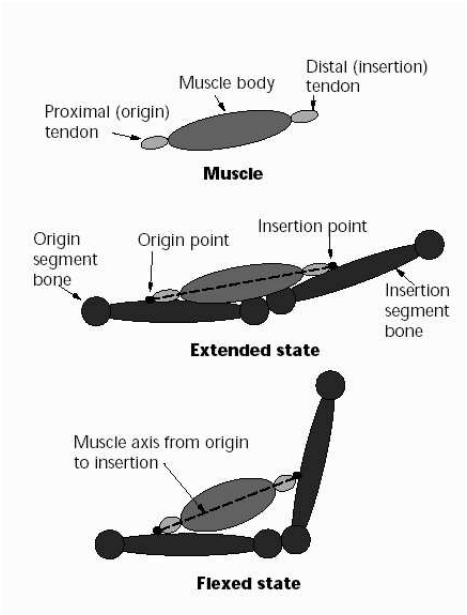


FIG. 3.9 – Illustration de la contraction sur le modèle de muscle de Wilhelms [Wil97]. On peut noter que le muscle est constitué de 3 ellipsoïdes : un central qui représente le muscle et un à chaque extrémité qui joue le rôle de tendon.

$l = 2c$, la longueur de l'ellipsoïde, $2a$ et $2b$ les longueurs des deux autres axes principaux. Soit v, le volume que l'on veut garder constant : $v = (4*\pi*a*b*c)/3$. Soit r, le rapport $\frac{a}{b}$ que l'on souhaite aussi garder constant. En effet, lorsque le muscle gonfle, on souhaite le voir gonfler de la même manière dans toutes les directions. On a alors $b^2 = \frac{3v}{4\pi*r*c}$. Ainsi, si l change, on a une nouvelle valeur de c et il est aisé de trouver la nouvelle valeur de b (donc de a) traduisant la contraction.

– Cas d'une contraction isométrique [SPCM97]
  Dans ce cas, les os ne bougent pas. Seul le muscle devient plus *tendu*. Les auteurs introduisent donc un nouveau paramètre, la tension t, pour ajuster le rapport r, qui, cette fois, n'est plus constant.

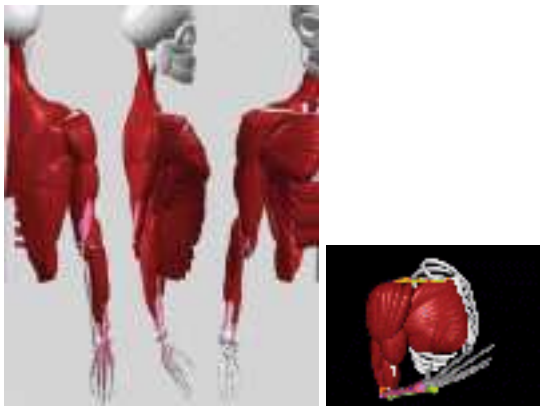


FIG. 3.10 – Muscles à base d'ellipsoïdes de Scheepers et al. [SPCM97].

Ces modèles ont été utilisés pour réaliser un torse et un bras d'humanoïde (voir figure 3.10). Le modèle général a été utilisé avec succès pour modéliser de larges muscles tels que les trapèzes ou le latissimus dorsi. Mais, si ce modèle simple est adapté pour les muscles fusiformes, et quelques muscles plats, il reste insuffisant pour beaucoup d'autres muscles.

Cylindre déformé

Dans sa première approche, Wilhelms [Wil97] utilise des ellipsoïdes pour représenter les muscles fusiformes. Ce modèle simple étant inadapté dans bien des cas, elle propose un second modèle [WG97] basé sur un cylindre déformé, essayant de fournir un bon compromis entre temps de calcul et réalisme.

Représentation

Le muscle est généré à partir d'un cylindre (voir figure 3.11) qui possède deux origines et deux insertions. Ces 4 points sont décrits par une position $3D$ sur des os spécifiques et paramétrés par la taille de la boîte englobante de l'os pour des raisons de réutilisation. Chaque muscle est un cylindre discrétisé et déformé : l'axe du cylindre est une courbe qui part du milieu des deux origines et aboutit au milieu des deux insertions. Ensuite, le cylindre est découpé en 7 tranches transversales à l'aide de 8 ellipses. Les deux ellipses des bouts englobent respectivement les origines et les insertions, et les 6 autres ellipses définissent la forme et l'axe longitudinal du muscle. Il n'y a pas de tendons. Chaque ellipse transversale est discrétisée en points espacés régulièrement et formant un polygone plan. Il suffit de connecter ces points avec leurs 4 voisins (deux sur la même ellipse, un sur l'ellipse précédente et un sur l'ellipse suivante) pour obtenir un maillage polygonal pour le rendu qui servira à calculer l'ancrage de la peau (voir section 3.5 pour plus de détails sur la déformation de la peau).

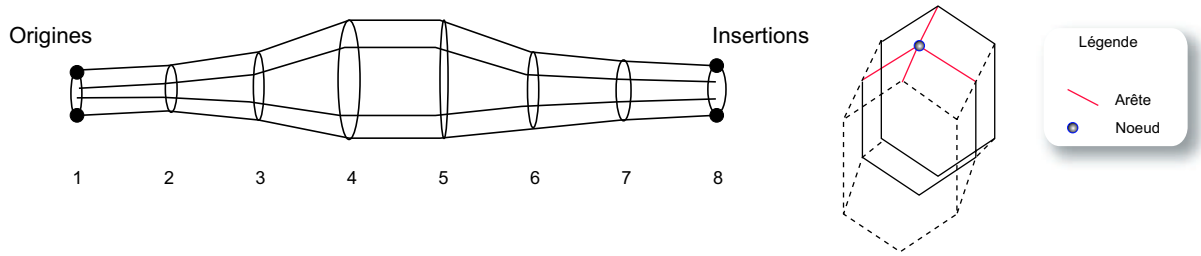


FIG. 3.11 – Modèle de muscle de Wilhelms, basé sur un cylindre déformé (image extraite de [WG97]).

L'enveloppe par défaut est créée automatiquement à partir des points origines et insertions et peut ensuite être modifiée interactivement par l'utilisateur en déplaçant des tranches (translation, rotation) ou en changeant la taille du muscle. L'utilisateur peut aussi ajouter un pivot. Dans ce cas, le muscle ne va pas des origines aux insertions, mais des origines au pivot puis du pivot aux insertions. Cela permet d'éviter des interpénétrations de matière (comme décrit dans la thèse de Nedel [Ned98]) et de simuler des muscles comme ceux de la cuisse qui passent sur le genou (le pivot est alors sur la rotule). Ce pivot peut être utilisé quel que soit le mouvement pour un muscle donné, ou bien seulement ponctuellement, pour quelques mouvements plus compliqués. Pour résumer, l'utilisateur peut contrôler les paramètres suivants : les origines, les insertions, le facteur d'échelle de l'enveloppe du muscle, la translation et la rotation de chaque tranche du muscle, l'existence ou non d'un pivot, son emplacement et son orientation.

Un avantage non négligeable de ce modèle est que tout est paramétré, donc, réutilisable. Par exemple, si l'on modélise un humain, il est possible de réutiliser le même modèle pour un autre

humain différent pour lequel on aura rallongé les bras, raccourci les jambes. Le modèle est le même, seul le fichier de paramètres doit être modifié.

Animation

L'animation du muscle implique le calcul automatique de sa nouvelle enveloppe chaque fois qu'une articulation positionnée entre les origines et les insertions du muscle bouge. Le mouvement de l'articulation implique un changement de position relative des origines et des insertions. Un nouveau muscle est donc recalculé en utilisant les mêmes paramètres utilisateur que l'ancien, puis, pour préserver un volume à peu près constant, la largeur et l'épaisseur du muscle sont ajustées. Soit la *longueur au repos*, la distance entre le milieu des origines, le pivot s'il existe et le milieu des insertions lorsque le muscle est au repos. Soit la *longueur courante*, la même distance calculée lorsque le muscle est dans la position courante. La largeur et l'épaisseur du muscle sont alors agrandies par $\sqrt{\frac{\text{longueur au repos}}{\text{longueur courante}}}$. Ceci résulte en un agrandissement des ellipses transversales si le muscle rétrécit et une diminution dans le cas contraire. Le volume est préservé exactement entre les régions de tranches parallèles et change légèrement dans les autres régions, les auteurs insistant sur le fait que la conservation de volume n'est pas biologiquement justifiée (contraction isométrique).

Ensuite, une peau élastique, décrite à la section 3.5 est ajoutée au modèle. Ce modèle très simple permet malheureusement de ne simuler que des muscles fusiformes, pour lesquels seul un mouvement statique a été calculé.

3.3.2 Modèles physiques

En dehors des modèles géométriques pour la modélisation des muscles présentés précédemment, on trouve des modèles physiques dont la modification ne dépend plus de paramètres totalement géométriques, mais est calculée à partir de l'utilisation de ressorts ou d'éléments finis par exemple.

Pump it up

Un des premiers travaux sur la modélisation de muscles fusiformes par modèle physique est celui de Chen et Zeltzer [CZ92]. Ils utilisent un modèle biomécanique de l'action du muscle pour appliquer des forces non linéaires à un modèle d'éléments finis. Ces forces agissent dans la direction longitudinale du muscle et permettent de déformer le modèle. Le muscle étant attaché aux os et l'une de ses extrémités étant fixée, la déformation du muscle provoque un mouvement de l'os, spécifié par cinématique inverse.

Le gros inconvénient de ce modèle est qu'il est basé sur des données biomécaniques mesurées sur de vrais muscles. Si cette technique a permis la modélisation d'un muscle fusiforme du bras, elle n'a pas été utilisée pour modéliser d'autres sortes de muscles. Ensuite, le principe même est contraire aux techniques utilisées couramment en animation où c'est le squelette qui est d'abord animé et les muscles déformés en conséquence. Cela rend donc cette méthode difficile à utiliser pour faire réaliser à un personnage un mouvement donné.

Masses ressorts

Le modèle le plus simple (au sens physique) est celui de Nedel [Ned98, NT98, NT00] qui utilise un réseau masses-ressorts surfacique pour modéliser chaque muscle.

Représentation du muscle de Nedel

Seuls les muscles fusiformes qui permettent d'articuler le squelette ont été modélisés par cette technique. Chaque muscle est représenté par une ou plusieurs *lignes d'action* et une enveloppe. On peut voir la *ligne d'action* comme le *squelette* du muscle qui est la ligne droite le long de laquelle agit la force musculaire. Une *ligne d'action* est dans le cas général une droite, mais peut aussi être une ligne brisée si l'on ajoute des points de contrôle, pour obliger le muscle à passer par un endroit particulier et éviter ainsi à certains muscles de passer au travers de l'os (et de raccourcir pendant une élongation). Le muscle ne couvre pas nécessairement toute la *ligne d'action*, les bouts restants représentant alors les tendons et gardant une longueur constante pendant la simulation.

La création de l'enveloppe peut être faite de plusieurs manières : à partir de données médicales, en utilisant des ellipsoïdes ou en dessinant le muscle à la main autour de la *ligne d'action*. Ensuite, l'enveloppe du muscle est découpée en tranches afin de créer un maillage surfacique, orienté parallèlement et perpendiculairement par rapport à la *ligne d'action*. Cette étape peut permettre de gérer des niveaux de détails car on peut choisir l'ordre de grandeur du nombre de points du maillage final. On procède donc de la manière suivante :

1. On découpe le muscle en n tranches perpendiculaires à la *ligne d'action*. Sur chacune de ces tranches, on choisit y points de la surface du muscle, intersections entre une ligne imaginaire partant du centre de la tranche (là où passe la *ligne d'action*) allant dans une direction définie et la surface du muscle. On obtient les $n * y$ points du maillage.
2. On ajoute des ressorts linéaires pour simuler l'élasticité du muscle : on met des ressorts *verticaux* entre les points se correspondant d'une tranche à l'autre et on met des ressorts *horizontaux* entre les points voisins sur une même tranche (voir schéma du haut de la figure 3.12). L'élasticité des ressorts horizontaux est différente de celle des ressorts verticaux.
3. On ajoute des ressorts angulaires pour empêcher le muscle de se tordre et conserver un volume *à peu près* constant. Pour chaque masse x, on considère les 4 voisins (les deux de la même tranche x_1, x_3, et les deux des deux tranches voisines de part et d'autre : x_2, x_4). Le premier ressort relie le milieu de x_1x_3 avec x et le second, le milieu de x_2x_4 avec x. Tous les ressorts sont placés lorsque le muscle est au repos (voir figure 3.12).
4. On ajoute des contraintes géométriques résolues par dynamique inverse. Elles peuvent être locales, zonales ou globales et servent essentiellement à éviter les interpénétrations de matière. Dans les derniers travaux publiés en 2000, les contraintes étaient utilisées uniquement pour bouger les points d'attache du muscle au squelette.

Le muscle étant modélisé par des ressorts, l'utilisateur doit définir 6 paramètres pour que les calculs soient possibles : 2 coefficients d'élasticité pour les ressorts linéaires horizontaux et verticaux, 2 coefficients de courbure pour les ressorts angulaires verticaux et horizontaux, la masse du muscle et un facteur d'amortissement. Si la masse du muscle est intuitive, les autres coefficients le sont moins, surtout pour un utilisateur peu familier avec les ressorts.

Déformation du muscle

Le mouvement est créé au niveau des articulations. Ce mouvement induit un mouvement de tout ce qui est indéformable, des os du squelette en particulier. Ceci résulte en un changement de la *ligne d'action* (les positions de l'origine, de l'insertion et des points de contrôle doivent être recalculées dans le repère de l'humanoïde). Concrètement, la *ligne d'action* s'agrandit ou rétrécit, éloignant ou rapprochant l'origine du muscle, l'insertion et les points de contrôle. Ceci entraîne des contraintes géométriques et un nouvel état d'équilibre doit être calculé à chaque pas de temps. Le mouvement est simulé en appliquant un système d'équations du mouvement à chaque particule du modèle. Soit

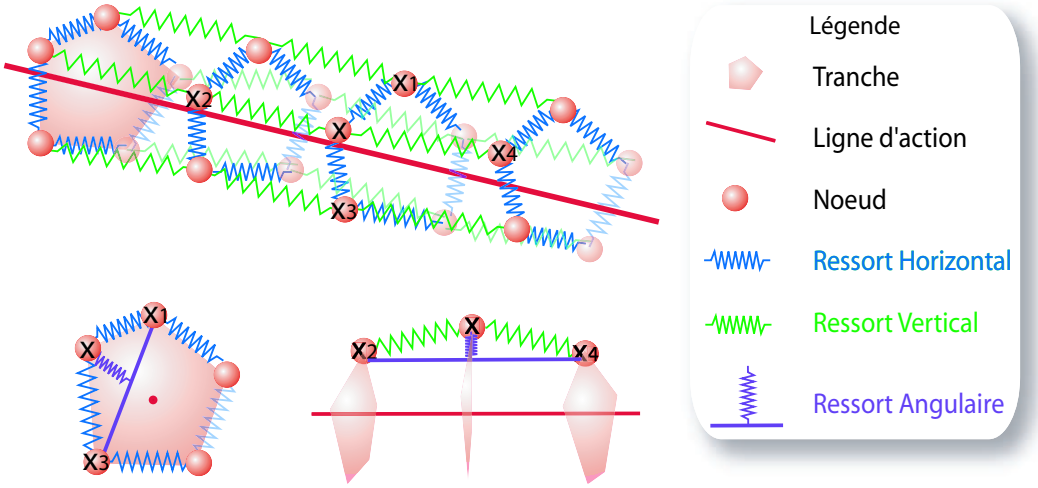


FIG. 3.12 – Description du maillage masses-ressorts du muscle de Nedel (image inspirée de [Ned98]).

$fresult_i$, la force résultante au point i. Elle est égale à la somme des forces dues aux ressorts linéaires et angulaires respectivement :

$$fresult_i = felasticity_i + fcurvature_i$$

Les équations de Lagrange (voir [TPBF87]) donnent :

$$m\ddot{xi} + \gamma\dot{xi} + fresult_i = fextern_i \tag{3.3}$$

avec $fextern_i$ nulle, et des contraintes sur certaines masses.

Ces équations sont ensuite intégrées dans le temps à l'aide de la méthode de Runge Kutta de degré 4 (voir section 2.3.2.3).

Le problème de cette méthode semble être tout d'abord l'instabilité numérique. Le pas de temps utilisé pour l'intégration doit être extrêmement faible pour que la simulation ne diverge pas. Ensuite, il faut probablement plusieurs itérations pour obtenir une forme correcte de l'enveloppe, ce qui fait que cette méthode ne peut pas être utilisée pour des effets dynamiques. De toute façon, les temps de calcul sont trop longs puisqu'un seul muscle pouvait être simulé en temps-réel en 2000.

Surface polygonale et ressorts

Dans le même ordre d'idée, Aubel [Aub02] présente un modèle de muscle qui couple une *ligne d'action* et une enveloppe surfacique. Cette fois, à l'inverse de Nedel [NT98, Ned98, NT00], les ressorts servent à déformer la *ligne d'action*, l'enveloppe étant une surface polygonale.

Les muscles sont modélisés individuellement, les muscles et leurs tendons étant représentés par une surface polygonale triangulée. Chaque muscle se compose de deux parties :

- le squelette du muscle défini par les *lignes d'action* introduites par Nedel [NT98] et
- un maillage surfacique qui représente l'enveloppe du muscle.

De la même façon que dans [NT98, Ned98, NT00], c'est la *ligne d'action* du muscle qui engendre les déformations et c'est donc sur cette *ligne d'action* que les efforts ont été concentrés.

Représentation de la ligne d'action du muscle

Aubel [Aub02] propose deux modèles de *lignes d'action* visant des buts différents : un modèle physique, lent mais précis et un modèle géométrique beaucoup plus rapide destiné à être utilisé en temps-réel.

Modèle physique

Dans ce modèle, la *ligne d'action* est constituée d'un ensemble de ressorts mis bout à bout. Les deux extrémités de cette chaîne sont attachées aux os et entre ces deux extrémités, on a un ensemble de nœuds dynamiques dont les positions résultent de la relaxation des ressorts (lorsque les deux extrémités se rapprochent). Tous les nœuds de la *ligne d'action* ont la même masse et la raideur de chaque ressort dépend de ce qu'il représente : tendon ou ventre du muscle. Cette *ligne d'action* est contrainte par l'ajout de champs de forces attractifs pour raffiner la trajectoire, répulsifs pour empêcher par exemple que les ressorts ne pénètrent dans les os. Enfin, pour éviter certains problèmes d'angles, il est possible d'approcher une partie de la *ligne d'action* par une courbe de Bézier cubique.

Modèle géométrique

Dans ce modèle, on enlève l'élasticité du modèle précédent. La *ligne d'action* devient donc un segment qui possède des nœuds dynamiques. Les positions de ces nœuds varient suivant la longueur de la *ligne d'action* ce qui entraîne des modifications du muscle. Ce modèle est beaucoup plus rapide, mais moins précis et peu adapté aux muscles complexes. D'autre part, puisque les forces ne sont pas modélisées, il n'est pas possible d'avoir un retour sur les forces qui entrent en jeu lors du mouvement.

Animation

L'animation de l'un ou l'autre modèle est simple. Lorsque les extrémités de la *ligne d'action* se rapprochent, les nœuds dynamiques bougent et une nouvelle enveloppe du muscle est calculée par une technique de type *Skeleton Subspace Deformation* (voir section 2.3.1.2). La *ligne d'action* joue le rôle du squelette. Cependant, ce modèle permet seulement d'obtenir des déformations statiques.

Pour ajouter des déformations dynamiques, Aubel ajoute des ressorts avec un fort amortissement qui entraînent une *ligne d'action* virtuelle. Chaque nœud de la *ligne d'action* est donc attaché par un ressort au nœud correspondant sur sa trajectoire cinématique. Ceci résulte en un mouvement dynamique de la *ligne d'action* et donc du muscle. La position finale de la *ligne d'action* retenue est, pour chaque nœud, une interpolation linéaire entre la position dynamique et la position cinématique. Avec cette technique, Aubel obtient de bons résultats visuels. Cependant, il ne précise pas comment il choisit les poids de son interpolation linéaire (probablement les mêmes pour tous les nœuds), ni comment il contraint ses ressorts à vibrer dans la même direction en même temps. Il utilise pour cela probablement des ressorts identiques pour tous les nœuds.

Sa technique est beaucoup moins coûteuse qu'un modèle physique complet (type éléments finis). Néanmoins, chaque muscle doit être modélisé individuellement et recouvert d'une couche de graisse et de peau, ce qui accroît rapidement les temps de calcul.

En bref : Un certain nombre de modèles de muscles ont été proposés dans la littérature. Tous ces modèles permettent de calculer des déformations cinématiques, en se basant sur des lois de déformation du muscle. Ainsi, quand un muscle fusiforme allonge, le rayon de son ventre diminue afin de conserver un volume constant. L'inconvénient majeur des modèles anatomiques est le temps qui est nécessaire à l'infographiste pour effectuer la modélisation. En effet, chaque muscle doit être modélisé pour obtenir des animations satisfaisantes. Pour les mêmes raisons, les temps d'animation sont souvent longs et loins du temps-réel ce qui rend ces techniques inutilisables dans les jeux vidéos.

3.4 Tissus

Lorsqu'un personnage se déplace, il subit deux types de déformations au niveau de son apparence externe : des déformations cinématiques dues aux muscles, et des mouvements dynamiques, en grande partie dûs à tous les autres tissus (adipeux entre autres). Selon l'article, le terme *tissus* englobe une ou plusieurs parties de l'anatomie. Il est le plus souvent utilisé pour tout ce qui n'est pas du muscle : graisse, organes internes, derme, ou bien pour désigner une couche mixte muscles-graisse. Comme nous l'avons vu, l'existence de lois cinématiques de déformation des muscles a permis de construire un certain nombre de modèles. Les tissus, eux, sont plutôt caractérisés par leur élasticité et leur capacité à vibrer. C'est pourquoi ils sont décrits par des modèles physiques, discrets ou continus.

3.4.1 Masses-ressorts

C'est la mécanique la plus simple pour effectuer des simulations physiques. Cependant, les réseaux masses-ressorts sont souvent difficiles à contrôler, et les intégrations numériques ne sont pas toujours stables. Leur avantage par rapport aux éléments finis est leur facilité de mise en œuvre et des coûts de calcul réduits.

Le premier modèle de déformations dynamiques a été proposé par Chadwick et al. [CHP89] en 1989 pour modéliser les joues d'un personnage. Afin de présenter son modèle physique, rappelons tout d'abord que dans son modèle géométrique, les déformations statiques sont calculées à l'aide de FFDs attachées par paires (voir section 3.3.1). Pour passer à des déformations dynamiques, il fait correspondre les cubes de la grille de contrôle FFD à des cubes masses-ressorts. Chaque point de contrôle devient une masse et ces masses sont toutes reliées entre elles par des ressorts au sein d'un cube (y compris en diagonale).

À chaque pas de temps, les forces dues aux ressorts sont calculées et appliquées aux particules qui répondent par une accélération dans la direction où la force a été appliquée. Les équations du mouvement sont intégrées en utilisant la méthode de Runge Kutta de degré 2 avec un pas de temps adaptatif. Bien sûr, certains points doivent être fixés, notamment ceux qui sont attachés aux os. La dynamique est appliquée en post-traitement, après le calcul du mouvement du squelette articulé. L'algorithme est le suivant :

1. Le mouvement complet du personnage est pré-calculé.
2. À ce moment là, les positions de chaque point de contrôle de la FFD sont connues. On en fixe certains, ceux qui sont collés aux os par exemple.
3. On construit le modèle physique (on met les ressorts) et le mouvement des points fixes est utilisé pour induire le mouvement des points dynamiques.

4. On calcule les nouvelles positions des points grâce aux équations du mouvement et les positions de ces points sont ensuite mises en correspondance avec les points de contrôle de la FFD, ce qui définit donc la forme déformée dynamiquement de la chair.

Il faut nécessairement choisir des constantes de raideur et d'amortissement des ressorts, et ceci est fait en accord avec l'apparence visuelle des résultats lors de l'animation. Ils ne sont pas nécessairement basés sur des réalités physiques.

L'avantage du réseau masses-ressorts couplé à des FFDs est qu'il est très facile de passer d'un modèle d'animation à l'autre, c'est à dire de passer d'une simulation cinématique à une simulation dynamique, puisque la représentation du modèle est déjà volumique.

3.4.2 Éléments finis

La quasi totalité des travaux sur la simulation cinématique ou dynamique de tissus emploie des éléments finis. Ceci se justifie par le fait que les simulations physiques basées sur les éléments finis donnent des résultats beaucoup plus proches de la réalité que les masses-ressorts, puisque la masse, la raideur et l'amortissement peuvent être définis de façon continue à l'intérieur d'un élément. Cependant, l'inconvénient principal de cette technique est la lenteur des calculs.

Éléments finis non temps-réel

Malgré cela, Gourret et al. [GTT89] utilisent des éléments finis pour modéliser une main qui saisit une balle. Les deux objets, la main et la balle, sont discrétisés en éléments finis cubiques (8 nœuds et 3 degrés de liberté par nœud). La modélisation des objets se fait à l'aide d'un système développé à l'époque à l'EPFL appelé SABRINA. Ce système permet de générer des surfaces triangulées pour les objets, de leur associer un maillage volumique, et de recréer des surfaces triangulées déformées pour ces objets si on lui fournit un maillage volumique déformé.

Le modèle d'objet utilisé est un modèle en couches constitué d'os géométriquement réalistes et de chair déformable. La chair volumique est discrétisée en éléments finis mais le maillage n'est pas uniforme. Il est plus dense dans les régions les plus sensibles car ces régions sont plus riches en informations.

Afin de calculer les déformations statiques des deux objets, Gourret et al. utilisent l'équation d'équilibre classique de l'élasticité linéaire (voir équation (2.9)). Cependant, étant donné que l'animateur ne peut qu'imposer un déplacement de la main mais pas une force, les calculs sont faits à partir du déplacement U_k imposé par l'animateur aux os de la main. Si l'on découpe le système matriciel en déplacements connus U_k, inconnus, U_u, forces connues F_k et forces inconnues F_u, l'équation (2.9) s'écrit de la manière suivant :

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \cdot \begin{bmatrix} U_u \\ U_k \end{bmatrix} = \begin{bmatrix} F_k \\ F_u \end{bmatrix} \tag{3.4}$$

On en déduit deux relations qui vont nous permettre de résoudre le système global composé de la balle et de la main itérativement :

$$K_{11}.U_u = F_k - K_{12}.U_k \tag{3.5}$$

$$F_u = K_{21}.U_u + K_{22}.U_k \tag{3.6}$$

La première étape consiste à placer la balle entre les doigts, quel que soit le matériau utilisé pour la balle. Ensuite, l'animateur place les doigts à l'intérieur de la balle et l'algorithme commence une série

d'itérations qui calcule et décide si cette position est valide ou non. Dès que la balle et la main sont en contact, elles sont fusionnées en un seul objet. La balle doit alors être repoussée pour empêcher le recouvrement. La suppression du recouvrement crée des forces réactives sur la surface de la balle qui sont appliquées à la peau. À l'équilibre, ces forces maintiennent des déplacements compatibles entre les surfaces des deux objets.
L'algorithme est le suivant :

1. Mouvement des articulations par une méthode classique (paramétrique, cinématique ou dynamique) et calcul des déplacements U_k des os.
2. Déplacer la peau selon U_k, sans la déformer. La main se retrouve donc à l'intérieur de la balle. Application des déplacements U_k sur les nœuds de la balle pour empêcher l'interpénétration. La balle est alors écrasée plus qu'il ne faut et la main est inchangée.
3. Résoudre les équations (3.5) et (3.6) pour obtenir les déplacements U_u sur la balle et les forces réactives F_u sur les points de contact de la balle.
4. Appliquer la force $-F_u$ sur les points de la peau pour garantir l'égalité et l'opposition des forces entre la balle et la peau à l'équilibre.
5. Faire comme si les forces de contact F_u étaient les forces prescrites F_k de la balle sur la peau.
6. Résoudre les équations (3.5) et (3.6) pour la main et la balle simultanément pour obtenir les déplacements U_u de la balle et de la peau, et les forces réactives F_u sur les os. Si l'équilibre n'est pas atteint et si les forces réactives sur les os ne dépassent pas un certain seuil de validité, on continue à l'étape 2, sinon, on s'arrête.

Le traitement global présenté ici peut aussi être appliqué à l'inter-déformation de doigts, à la déformation entre les doigts et la paume, c'est à dire pour tous les cas de compression et de dilatation entre deux morceaux d'humanoïde.

Le principal inconvénient de cet algorithme est que le nombre d'itérations n'est pas prévisible, ni garanti inférieur à une certaine limite. Cela compromet donc fortement le temps-réel. Un autre problème est que, même après un certain nombre d'itérations, on n'est pas sûr d'obtenir un résultat, seulement la certitude que la valeur du déplacement imposé au début à la main n'est pas bonne. Il faut alors recommencer une autre simulation. Le résultat n'est donc pas garanti non plus, ce qui est extrêmement gênant dans le cas d'un système interactif. Bien sûr, ces incertitudes se compensent par une très bonne modélisation des contacts entre les deux objets et de très bons résultats visuels.

Éléments finis temps-réel pour la modélisation d'un foie

Plus récemment, Cotin et al. [CDA98] utilisent les éléments finis pour la modélisation d'un foie. Leur but est de l'animer en temps-réel, malgré les temps de calcul énormes dûs à l'utilisation des éléments finis. Le modèle de foie doit servir à la simulation d'une opération avec un système à retour d'efforts. La simulation de déformations avec retour d'efforts se fait par une boucle fermée :

1. Le modèle se déforme en accord avec le mouvement induit par le chirurgien.
2. On calcule la force de contact à partir de cette déformation.
3. On génère cette force sur le système à retour d'efforts.

Le problème principal de ce genre de simulation est le temps-réel. Pour assurer un réalisme suffisant, le calcul des forces doit être effectué à une fréquence d'au moins 300 Hz, et le rendu à 24 Hz. Cependant, les méthodes de calcul basées sur les éléments finis ou les réseaux masses-ressorts sont trop coûteux en temps. Cotin et al. utilisent donc un algorithme particulier détaillé plus loin.

Élasticité linéaire

D'après [Fun93], le meilleur modèle pour le réalisme des tissus déformables est un modèle visco-élastique non-linéaire. Le problème est que c'est très long et très complexe à calculer. Comme première approximation de la loi de déformation de tissus déformables, on peut prendre une élasticité linéaire $3D$. Les déformations sont alors décrites par l'équation (2.9).

Lors de la simulation, des forces extérieures sont appliquées sur la surface du solide alors que quelques nœuds sont fixés. Étant donné que l'on utilise un système à retour d'efforts, il est impossible de mesurer la force exercée par le manipulateur. La seule chose que l'on connaît est la position de l'extrémité de l'outil qui appuie sur le solide. Les calculs de déformation doivent donc se faire avec des contraintes sur le déplacement U et non sur la force F. Les conditions aux limites sont principalement les déplacements des points de contact entre le corps déformable et l'instrument chirurgical. On peut donc déduire la force exercée sur le capteur situé au bout de l'instrument et la déformation globale du corps par application des équations de l'élasticité linéaire. Les forces calculées peuvent ensuite être retournées au système à retour d'efforts pour opposer une résistance au mouvement de la main du chirurgien. Pour inclure les nouvelles contraintes, on remplace le système $F = KU$ par le système suivant :

$$\begin{bmatrix} K & \bar{K} \\ \bar{K}^T & 0 \end{bmatrix} \begin{bmatrix} U \\ \lambda \end{bmatrix} = \begin{bmatrix} F \\ U^* \end{bmatrix} \tag{3.7}$$

Ce système provient de la méthode des multiplicateurs de Lagrange utilisée pour imposer certaines valeurs aux variables solution. Les valeurs λ_i de λ obtenues après résolution du système sont égales à l'opposé de la force qui doit être appliquée au degré de liberté u_i pour imposer le déplacement u_i*. Les résultats obtenus de cette façon (élasticité linéaire) sont uniquement valables pour des petits déplacements, ce qui convient très bien dans le cas de cette simulation car les déplacements sont limités à cause du système à retour d'efforts. L'intérêt de l'élasticité linéaire est la possibilité de calculer n'importe quelle déformation à partir de la connaissance d'un nombre fini de déformations élémentaires. L'autre intérêt est qu'il est possible de pré-calculer la matrice de raideur K et de ne l'inverser qu'une seule fois. Le gros problème de cette approximation est que ces équations ne sont plus du tout valables dans le cas de grands déplacements. On observe alors de grandes variations de volume, complètement irréalistes.

Élasticité non-linéaire

Dans une seconde étape, le modèle de Cotin et al. [CDA98] a été amélioré pour simuler une élasticité quasi non-linéaire. Imaginons un cylindre sur lequel on applique une force, dans la direction de son axe. Soit x, le déplacement axial produit, soit $f(x)$ la courbe décrivant la force appliquée au modèle et soit $dr(x)$ la courbe décrivant le déplacement radial résultant. À l'aide d'une minimisation aux moindres carrés, on peut approcher ces fonctions par deux polynômes : respectivement $Q(x)$ et $P(x)$, de degré compris entre 4 et 10 suivant la précision désirée. Pour étendre ces résultats au cas $3D$ tout en préservant une loi de déformation basée sur l'élasticité linéaire, on fait les propositions suivantes :

- Le calcul d'une déformation quasi non-linéaire sera basé sur un résultat de déformation linéaire, utilisant le même maillage avec les mêmes contraintes. En conséquence, $dr(x)$ et $f(x)$ doivent être exprimées en tant que fonctions du déplacement linéaire $dr_l(x)$ et $f_l(x)$.
- La variation en amplitude du déplacement radial doit être similaire à celle des courbes expérimentales. Le déplacement u_n associé au nœud n est décomposé en deux vecteurs : u_{n_t} et u_{n_r}, un colinéaire et un orthogonal à la direction de la contrainte (nécessaire car les résultats expéri-

mentaux ne donnent des renseignements sur la déformation radiale que dans le cadre de l'application d'une contrainte axiale sur le dessus de l'échantillon cylindrique). En conséquence, seule la variation non-linéaire de l'amplitude de la composante orthogonale est connue. Ainsi, l'amplitude de u_{n_r} est modifiée pour suivre la courbe $P(x)$ tandis que u_{n_t} continue d'avoir une variation linéaire. Pour exprimer $dr(x)$ et $f(x)$ en fonction respectivement de $dr_l(x)$ et de $f_l(x)$, on utilise le module d'Young E et le coefficient de Poisson ν établis empiriquement pour le matériau considéré. Pour un cylindre de hauteur h et de rayon r, la déformation radiale théorique est $dr_l(x) = \nu r x/h$ et $f(x) = sEx/h$ avec s, surface du haut du cylindre. De là, on peut déduire les fonctions $P(dr_l(x))$ et $Q(f_l(x))$ pour des propriétés élastiques particulières.

Considérons un nœud n avec le déplacement linéaire suivant : $u_n = u_n^t + u_n^r$. Le déplacement non-linéaire correspondant est $U_n = u_n^t + U_n^r$ avec

$$U_n^r = P(||u_n^r||)\frac{u_n^r}{||u_n^r||} \tag{3.8}$$

L'amplitude des forces est modifiée de la même façon, sauf que c'est la composante colinéaire qui est modifiée en accord avec $Q(x)$. La force non-linéaire au nœud n est donc $F_n = F_n^t + f_n^r$ avec

$$F_n^t = Q(||f_n^t||)\frac{f_n^t}{||f_n^t||} \tag{3.9}$$

Le modèle de déformation de Cotin est donc beaucoup plus élaboré que celui de Gourret qui n'était qu'une simple élasticité linéaire. Cotin présente un modèle d'élasticité non-linéaire qu'il est capable d'animer en temps-réel grâce à des pré-calculs judicieux.

Vers le temps-réel

Le meilleur moyen pour réduire les temps de calcul de déformation d'un modèle lorsqu'il s'agit d'un réseau masses-ressorts est bien évidemment de réduire le nombre de ressorts. C'est ce que fait Nedel [Ned98, NT98, NT00] par rapport à Chadwick [CHP89] en modélisant un muscle par un réseau de ressorts surfacique au lieu de volumique. Seulement, même si elle obtient du temps-réel pour un seul muscle, elle est vraisemblablement encore loin du temps-réel pour un humanoïde complet.

Une meilleure idée est probablement d'essayer de pré-calculer un maximum de choses hors ligne. C'est ce que font Cotin et al. [CDA98]. Ils basent leur algorithme sur le principe de linéarité et de superposition qui sont des conditions nécessaires.

A. Pré-calcul des déformations élémentaires

1. Spécification d'un ensemble de nœuds qui restent immobiles pendant la déformation. On assigne alors le déplacement $u_i^* = 0$ à chaque degré de liberté qu'il est nécessaire de spécifier, ce qui élimine des équations de la relation (2.9).
2. Pour chaque nœud libre k sur la surface du maillage, on impose un déplacement élémentaire $u_k^* = [\delta_1, \delta_2, \delta_3]^T$ et on calcule deux choses :
   - le déplacement de chaque nœud libre n différent de k et on le stocke dans un ensemble de tenseurs 3*3 $[T_{nk}^u]$ exprimant le déplacement du nœud n dû au déplacement élémentaire du nœud k imposé ;
   - les composantes de la force élémentaire λ au nœud k qui sont stockées dans le tenseur 3*3 $[T_k^f]$ et calculées à l'aide du système (3.7).

Cette étape de pré-calculs peut durer de quelques minutes à plusieurs heures. Les résultats sont stockés dans un fichier.

B. Déformations en temps-réel avec une élasticité linéaire
Le déplacement u_n d'un nœud n induit par la contrainte u_k^* appliquée au nœud k peut être obtenu par l'équation linéaire suivante :

$$u_n = [T_{nk}^u]\frac{u_k^*}{||u_k^*||} \tag{3.10}$$

pour chaque nœud n différent de k.

En général, plusieurs nœuds sont déplacés sur la surface du corps, donc, le déplacement d'un nœud est la somme de tous les déplacements induits par un nœud contrôlé. Cependant, la somme n'est pas directement applicable, car les nœuds contrôlés s'influencent mutuellement. On peut donc calculer la contrainte $\tilde{u}_{k_l}^*$, $l = 1, ..., m$, apportée par chaque nœud à la somme à partir des précédents tenseurs de déformation $[T_{nk}^u]$. La matrice suivante représente l'influence mutuelle de tous les nœuds contrôlés :

$$\begin{bmatrix} M \end{bmatrix} = \begin{bmatrix} \begin{bmatrix}\delta\end{bmatrix} & \begin{bmatrix}T_{21}^u\end{bmatrix} & \begin{bmatrix}T_{31}^u\end{bmatrix} & \cdots & \begin{bmatrix}T_{m1}^u\end{bmatrix} \\ \begin{bmatrix}T_{12}^u\end{bmatrix} & \begin{bmatrix}\delta\end{bmatrix} & \begin{bmatrix}T_{32}^u\end{bmatrix} & \cdots & \begin{bmatrix}T_{m2}^u\end{bmatrix} \\ \vdots & & & & \vdots \\ \begin{bmatrix}T_{1m}^u\end{bmatrix} & \begin{bmatrix}T_{2m}^u\end{bmatrix} & \begin{bmatrix}T_{3m}^u\end{bmatrix} & \cdots & \begin{bmatrix}\delta\end{bmatrix} \end{bmatrix} \tag{3.11}$$

avec

$$\begin{bmatrix}\delta\end{bmatrix} = \begin{bmatrix} \delta_1 & 0 & 0 \\ 0 & \delta_2 & 0 \\ 0 & 0 & \delta_3 \end{bmatrix} \tag{3.12}$$

Le vecteur de contraintes modifié $[\tilde{u}^*]$ qui doit être appliqué au maillage est $[\tilde{u}^*] = [M]^{-1}[u^*]$ Puis, grâce au principe de superposition, on calcule le déplacement total de chaque nœud n :

$$u_n = \sum_{l=1}^{m}[T_{nk_l}^u][\tilde{u}_{k_l}^*]$$

La force associée à un nœud contrôlé k est déterminée par : $f_k = [T_{nk}^f][u^*]$ (c'est la force qu'il faut appliquer au nœud k pour produire le déplacement $\tilde{u}^*$).

Pour les déformations quasi non-linéaires en temps-réel, on se sert des tenseurs $[T_{nk}^u]$ et $[T_{nk}^f]$ pré-calculés pour des déformations linéaires. Ils nous permettent de calculer u_n et f_n. On peut donc en déduire u_n^r, f_n^r et u_n^t, f_n^t, les composants colinéaires et orthogonaux et appliquer les relations (3.8) et (3.9), peu coûteuses en temps de calcul.

Le principal problème des éléments finis est le coût prohibitif des calculs. En effet, la quantité de calculs est liée à deux choses : tout d'abord, elle est directement proportionnelle au nombre d'éléments du maillage, et, par ailleurs, elle dépend fortement des lois de l'élasticité choisies. Cotin parvient, grâce à de lourds pré-calculs et l'utilisation de la loi de comportement du matériau simulé, à animer des déformations élastiques non-linéaires en temps-réel. Cependant, toutes les méthodes décrites précédemment ne permettent de simuler que des déformations statiques, c'est à dire sans dynamique. Les techniques suivantes relèvent donc le défi de simuler des déformations dynamiques, tout en gardant le temps-réel. Pour cela, certains utilisent des approches *hiérarchiques* afin de pouvoir simuler un grand nombre d'éléments en temps-réel.

Éléments finis hiérarchiques adaptatifs

Debunne et al. [Deb00, DDCB01] proposent d'utiliser des éléments finis adaptatifs afin de ne pas perdre de temps à simuler des zones de l'objet qui ne subissent pas de déformations. Lorsque l'on souhaite des déformations précises sur un modèle, il est évidemment nécessaire d'avoir un maillage fin. Le problème est que quand on affine un maillage, le nombre d'éléments croit très vite. Pour contrecarrer ce problème et donc améliorer les temps de calcul, Debunne utilise des éléments finis hiérarchiques. Il s'agit en fait d'avoir comme représentation du modèle différents maillages à différentes résolutions. Ensuite, lors de la simulation, on active les nœuds les plus fins dans les zones où les déformations sont les plus importantes. Ce raffinement est adaptatif et se base sur un seuil qui permet de juger si l'approximation linéaire courante de la déformation est acceptable ou non. Au moment d'un raffinement, un nœud grossier est désactivé et les nœuds du niveau plus fin qui doivent être alors animés sont choisis comme les plus proches du nœud grossier en se basant sur une partition de l'espace en régions de Voronoï.

La résolution du système des éléments finis utilise un tenseur des déformations adapté et prend en compte l'amortissement du matériau. Afin d'obtenir une solution au système en temps-réel, les fonctions de base des éléments sont choisies linéaires. Enfin, le système est résolu en traitant chaque élément indépendamment à l'aide d'une approximation locale, et avec un pas de temps adaptatif garantissant un taux minimal de rafraîchissement des images.

Cette technique permet donc de simuler des déformations dynamiques pouvant comporter de grands déplacements, en temps-réel. La contrainte la plus gênante est qu'il faut avoir en même temps la représentation volumique du modèle à tous les niveaux de détails nécessaires, ce qui peut être légèrement coûteux en place mémoire. Cette représentation pourrait aussi poser problème dans le cas de fracture du maillage lorsque celle-ci se produit à un niveau plus fin que le niveau grossier (comment couper l'élément grossier dans ce cas ?).

Base hiérarchique de fonctions

Plus récemment, Capell et al. [CGC +02a] utilisent un modèle FFD hiérarchique. L'objet est plongé dans une grille de contrôle facilement paramétrable qui peut se raffiner par subdivision (se reporter à la section 2.3.1.3 pour plus de détails) et sur laquelle une base hiérarchique est définie. Chaque élément de la grille est alors un élément fini. Au moment de la simulation, sont actifs les éléments les plus fins dans les zones où plus de détails sont nécessaires.

La dynamique du modèle est modélisée par éléments finis appliqués à la formulation de Lagrange des équations de l'élasticité. Dans leur second papier, Capell et al. [CGC +02b] ajoutent un squelette en fixant certains éléments qui ne se déforment plus mais entraînent la déformation du reste. Pour cela, le squelette est choisi pour coïncider avec les arêtes des éléments.

Stiffness Warping

La méthode présentée par Müller et al. [MDM +02] traite plus particulièrement du problème des larges déformations en élasticité non-linéaire. Le problème principal de l'élasticité linéaire est que les forces élastiques linéaires sont invariantes par translation, mais pas par rotation, ce qui fait que le volume de l'objet n'est pas conservé lors de larges déformations. L'inconvénient majeur de l'élasticité non-linéaire pour obtenir des temps de calcul faibles est que la matrice de raideur de l'objet change à chaque pas de temps et qu'il n'est donc pas possible de la pré-calculer.

La méthode de Müller résout ces deux problèmes car les calculs se font à partir de la matrice

d'élasticité linéaire pré-calculée qui est ensuite transformée par un champ de tenseurs qui décrit les rotations locales du matériau déformé. Il s'agit donc d'extraire la partie rotationnelle de la déformation de la matrice de raideur. De cette manière, l'objet garde un volume constant et il est possible de pré-calculer la matrice de raideur. Pour cela, à chaque pas de temps, on calcule le champ de tenseurs qui décrit les rotations locales de tous les sommets du maillage.

En élasticité linéaire, les forces élastiques pour un tétraèdre sont :

$$f_{elastic} = K(x - x_0) \tag{3.13}$$

avec K, la matrice de raideur de l'élément,
$f_{elastic}$, les forces élastiques,
$x \in \mathbb{R}^{12}$, les positions déplacées des 4 sommets du tétraèdre,
et $x_0 \in \mathbb{R}^{12}$, les positions originales des 4 sommets du tétraèdre.

D'autre part, soit $R_x \in \mathbb{R}^{3*3}$, une matrice de rotation $3D$ orthogonale représentant une composante rotationnelle globale de la transformation rigide du tétraèdre. Soit $R_e \in \mathbb{R}^{12*12}$, qui contient 4 copies de R_x sur sa diagonale et 0 partout ailleurs :

$$R_e = \begin{bmatrix} R_x & 0 & 0 & 0 \\ 0 & R_x & 0 & 0 \\ 0 & 0 & R_x & 0 \\ 0 & 0 & 0 & R_x \end{bmatrix} \tag{3.14}$$

Étant donné que l'on connaît la rotation des 4 sommets du tétraèdre (R_e), on peut enlever cette rotation, ($R_e^{-1}x$), calculer la force élastique en élasticité linéaire sans la rotation (dans ce cas, la formule est valable) et remettre la rotation ensuite. Cela donne mathématiquement :

$$f_{elastic} = R_e K(R_e^{-1}x - x_0) \tag{3.15}$$

Il suffit donc d'appliquer ce calcul de force à chaque sommet de chaque tétraèdre. Cette technique fonctionne en temps-réel et les résultats obtenus sont très proches de ceux obtenus par une vraie élasticité non-linéaire, mais à moindre coût. Cette méthode peut aussi bien être appliquée à des éléments finis qu'à des réseaux masses-ressorts. Elle a été appliquée à un chien qui secoue la tête. Seule la tête du chien a été simulée, mais les résultats sont impressionnants.

Dynamic Response Texture

James et Pai [JP02] ont développé un modèle physique capable de simuler des déformations dynamiques d'un objet volumique à la géométrie complexe en temps-réel. Ce modèle, appelé DyRT pour *Dynamic Response Texture*, peut être rajouté à n'importe quel modèle d'animation conventionnel et est traité au même moment et de la même façon que le rendu en utilisant les capacités de calcul de la carte graphique.

Les déformations de la DyRT sont basées sur l'analyse modale. Le modèle d'humanoïde utilisé est constitué d'os et de chair. Les os sont responsables du mouvement et de l'excitation des modes vibratoires des DyRT qui y sont rattachées. Pour les parties du corps déformables nécessitant du dynamique, on crée un maillage volumique d'éléments finis tétraédriques. Ensuite, pour chaque maillage, on calcule les modes de vibration [PW89] et on les enregistre dans la mémoire de la carte graphique.

L'algorithme de construction du modèle est le suivant :

1. Créer la géométrie du personnage (os articulés et chair).
2. Pour chaque partie déformable, on construit un objet DyRT :
   - on définit un volume élastique ;
   - on génère un maillage de ce volume ;
   - on fixe certains éléments du maillage (les sommets collés aux os) ;
   - on définit les propriétés matérielles de ce volume : raideur, compressibilité et densité ;
   - on calcule les modes dominants par un logiciel d'analyse modale (les m fréquences modales λ_i et les m enveloppes volumétriques ϕ_i) ;
   - on définit les m cartes de perturbation des normales N_i, calculées à partir de la géométrie du personnage ;
   - on calcule les filtres de convolution et la matrice de transfert qui servent à induire les vibrations à partir du mouvement (rotation, translation) de l'os du squelette auquel est rattaché l'élément de chair.

L'idée intéressante de cet article est de ne rajouter les effets dynamiques que là où ils sont nécessaires au lieu de créer un modèle élastique complet d'un humanoïde par exemple. L'avantage est double : premièrement, en travaillant sur des maillages plus petits, les temps de calcul s'en trouvent réduits. Dans cette technique d'ailleurs, ce ne sont que des pré-calculs servant à l'analyse modale. Deuxièmement, puisque les maillages sont séparés pour chaque partie du corps (ce sont des objets DyRT différents), il est alors possible de donner des propriétés mécaniques différentes à ces différents maillages. On peut donc imaginer des muscles plus raides au niveau des cuisses et un ventre plus mou car il s'agit en réalité de tissus adipeux. De même, il est possible pour les mêmes raisons d'empiler plusieurs objets DyRT et ainsi de réaliser des couches de différentes propriétés mécaniques. Cependant, cela n'a pas été réalisé par les auteurs. Enfin, le temps-réel dans cette technique est surtout dû à l'utilisation de l'analyse modale. La carte graphique ne joue finalement qu'un rôle mineur mais essentiel : la superposition linéaire des modes retenus point par point.

En bref : Les tissus déformables sont bien souvent modélisés par des éléments finis, cette approximation spatiale et temporelle étant la plus exacte. Afin d'améliorer les temps de calcul, il est possible d'effectuer des pré-calculs, d'utiliser une représentation hiérarchique pour ne simuler que ce qui est nécessaire ou bien d'avoir recours à des techniques d'analyse modale. L'état de l'art actuel propose une variété de solutions satisfaisantes. Néanmoins, ce genre de techniques est difficilement utilisable par un infographiste, qui ne possède qu'un modèle surfacique de son personnage. Par rapport aux buts que nous nous sommes fixés, à savoir rajouter des effets dynamiques à une animation *existante*, le papier qui satisfait au mieux nos contraintes est celui de James et Pai [JP02] qui utilise des éléments de chair animés en temps-réel par analyse modale. Cependant, cette technique requiert un certain nombre de pré-calculs qui ne sont pas à la portée de l'infographiste. Elle requiert aussi une description volumique de certaines parties du personnage et une carte graphique adaptée, ce qui rend son utilisation difficile dans les jeux vidéos.

3.5 Peau élastique

Le plus souvent, les modèles en couches ne sont pas recouverts d'une peau élastique, celle-ci étant fusionnée avec l'enveloppe externe des muscles ou des tissus. Cependant, quelques papiers ont proposé soit une peau polygonale élastique modélisée par des ressorts surfaciques, soit une peau

calculée à partir d'une surface implicite *enveloppant* tous les composants internes et simulant une épaisseur de graisse.

3.5.1 Surface élastique

Turner et al. [TT93] proposent de modéliser la peau par une surface élastique continue. La surface est un maillage polygonal rectangulaire dont les sommets possèdent des attributs physiques tels qu'une masse, une élasticité, une raideur, ainsi que des informations de l'état courant comme une position ou une vitesse.

Cette surface élastique subit des forces répulsives et attractives. Les forces répulsives permettent de simuler la couche de graisse qui se trouve entre la peau et les muscles et qui tend à repousser la peau. Des ressorts attachés à certains muscles, quant à eux, tendent à attirer la surface.

Toutes ces forces ainsi que les propriétés de la peau sont prises en compte dans un solveur numérique [TPBF87] qui évolue à un pas de temps donné.

Cette technique permet de simuler des effets dynamiques de la peau, tout en contrôlant sa raideur, sa façon de glisser sur les muscles. Un autre avantage de cette technique est que la simulation de la peau élastique peut se faire à plusieurs résolutions, sur un maillage grossier pour des applications interactives, et sur un maillage raffiné pour des applications demandant plus de réalisme ou des vues rapprochées.

3.5.2 Surface triangulée

Wilhelms et Van Gelder [WG97] proposent une peau élastique discrétisée reliée aux os et aux muscles sous-jacents. La peau est une surface triangulée générée lorsque l'animal est dans sa position de repos. Elle est constituée de ressorts et attachée en chaque sommet aux muscles, os ou autres tissus, de manière non fixe pour lui permettre de glisser. Les points d'attache, appelés *ancres*, se déplacent en suivant les muscles et leurs déformations (compression, étirement) grâce à un paramétrage de leur position sur la primitive à laquelle ils sont rattachés.

Une fois le maillage triangulaire de la peau défini et relié aux composants sous-cutanés, on modélise chaque arête de la surface triangulée par un ressort dont la raideur k_e est :

$$k_e(v_1, v_2) = (a_1 + a_2)/len^2$$

avec v_1 et v_2 les deux extrémités de l'arête,
len, la longueur de cette arête au repos,
et a_1 et a_2, les aires des deux triangles ayant l'arête en commun. Cette formule permet de donner une meilleure uniformité des raideurs (mieux qu'une raideur constante appliquée à toutes les arêtes).

De même, les liaisons entre les ancres et leurs sommets correspondants du maillage de la peau sont modélisées par des ressorts de raideur k_a :

$$k_a = C_a . \sum_i (\frac{a_i}{3})$$

avec a_i, l'aire du triangle i ayant le sommet courant comme sommet (la somme se fait sur tous les triangles), et C_a, un facteur d'échelle permettant de faire plus ou moins glisser la peau sur les muscles (entraîne une raideur contrôlable mais qui reste proportionnelle à l'aire des triangles). La longueur à vide de ces ressorts est nulle.

Au moment de l'animation, lorsque le muscle se déforme, on calcule la nouvelle position des ancres dépendant du muscle, puis on applique un algorithme itératif de relaxation pour ajuster les sommets du maillage de la peau jusqu'à ce que les forces élastiques de la membrane atteignent l'équilibre. En effet, les points des couches inférieures auxquels sont attachés les sommets de la peau ont bougé. Les ressorts ont donc été tirés et ils doivent maintenant retrouver un état d'équilibre avant l'affichage.

L'algorithme de relaxation est le suivant :

1. On calcule le vecteur de force élastique pour chaque sommet. Soit $w_j = v_1 - v_2$, le vecteur reliant v_1, le sommet courant, à v_2, le sommet situé à l'autre bout de l'arête. La longueur de w_j est la longueur de l'arête. On en déduit le *length excess* qui est la différence de longueur entre la longueur actuelle et la longueur au repos. Bien sûr, cette longueur peut être négative.
2. On calcule la force élastique sur chaque arête en multipliant le *length excess* par la raideur k_e (qui peut avoir été modifiée selon le *length excess* pour favoriser le retour à la position d'équilibre par exemple). La direction de cette force est celle de w_j, c'est à dire qu'elle pointe vers v_1 si le *length excess* est positif, vers v_2 sinon.
3. Pour chaque sommet, on calcule la somme des forces élastiques dues à chacune des arêtes aboutissant à ce sommet et de la force due à l'arête reliant le point du maillage à son ancre. Ceci nous donne la force élastique agissant sur le sommet. Cette force est divisée par la somme des raideurs des ressorts contribuant à cette force pour donner le vecteur de relaxation élastique.
4. Tous les sommets de la peau sont translatés par le vecteur de relaxation élastique. Si l'on constate que ce vecteur fait entrer la peau en collision avec les composants des couches inférieures, on peut ajouter une force de rappel pour empêcher cette collision. Ensuite, on recommence à l'étape 1 tant que le vecteur de relaxation n'a pas atteint un seuil limite fixé par l'utilisateur ou un nombre maximum d'itérations.

Le point positif de cet algorithme de relaxation est que le nombre d'itérations est limité et l'on ne risque pas, comme dans le modèle de Gourret [GTT89], de voir la simulation s'éterniser. D'autre part, le résultat est garanti et visuellement très satisfaisant.

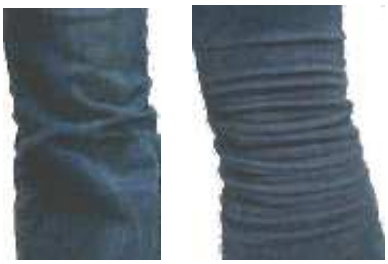
En bref : Dans beaucoup de modèles, comme celui de Scheepers [SPCM97], la peau est générée sous forme de surfaces implicites ce qui permet de régler l'épaisseur de graisse et de chair entre les muscles et la peau, mais ne va en aucun cas créer des effets de glissement ou des plis. Peu de papiers ce sont attachés à créer une peau élastique recouvrant les muscles, alors que c'est une couche très importante puisque finalement, c'est la seule visible à l'écran. Le problème est que les modèles anatomiques constitués de muscles individuels sont très gourmands en temps lors de leur utilisation. En effet, il faut modéliser et paramétrer les muscles un à un. De même, leur animation est souvent loin d'être temps-réel. On retrouve donc peu de modèles aboutis dans la littérature et la couche de peau est rarement modélisée.

3.6 Synthèse des détails externes : plis de la peau et/ou de vêtements

Même s'ils permettent de créer des animations de très bonne qualité, les modèles actuels pour l'animation de personnages, qu'ils soient surfaciques ou en couches, n'engendrent pas certains détails pourtant importants visuellement comme par exemple les plis. On peut en distinguer deux catégories :

les plis statiques, qui sont toujours présents et changent peu lors de l'animation et les plis dynamiques, qui se forment et se déforment au cours du temps. La première catégorie est le plus souvent créée lors de la modélisation, lorsque l'on se place dans le cas d'un *skinning* classique. L'infographiste incorpore directement ces plis dans le maillage ou dans la texture. Par contre, la seconde catégorie de plis a fait l'objet de beaucoup plus de recherches, car elle demande un travail énorme à un animateur, en l'absence de méthode spécifique.

Cette section s'intéresse au cas des rides qui ne sont pas fixes mais qui varient au cours de l'animation, suivant le mouvement de la surface (par exemple, le mouvement de la peau dictée par le mouvement d'un squelette). Quand le personnage bouge, la peau ou le vêtement glisse à la surface et des plis se créent aux endroits où la peau (ou le vêtement) s'accumule : aux articulations par exemple, comme on peut le voir sur la photo 3.13.)



(a) (b)

FIG. 3.13 – Plis réels sur l'avant (a) et l'arrière (b) d'un jean.

3.6.1 Bump et displacement mapping

Les premiers travaux sur les rides ont été réalisés par Blinn [Bli78] en 1978. Au lieu d'essayer de simuler les plis d'un maillage directement, il invente la technique du bosselage, plus connue sous le nom anglais de *bump mapping*. Il s'agit de modifier les normales de la surface avant d'effectuer les calculs d'éclairage en chaque point de la surface. C'est une technique simple, facilement implémentable sur les cartes graphiques actuelles, et qui est parfaitement adaptée à de petites perturbations de la surface. Cela permet d'obtenir des effets visuels de creux et de bosses sans que le maillage ne soit déformé. Cette technique a été reprise par plusieurs travaux [WKT97, BKMTK00, PP01] pour simuler des rides faciales. Le principal problème de cette méthode est que la silhouette de l'objet reste indéformée et on voit des défauts quand la caméra est très proche de l'objet (ou les plis trop gros) ou bien quand les rides se forment sur une surface très courbe comme les plis de pantalon au creux du genoux (voir figure 3.13 (b)). La seconde conséquence de la non déformation du maillage est que cela rend erronée la détection de collisions dans la région des plis. Cela pose problème surtout dans la simulation de vêtements où les plis sont suffisamment larges pour être impliqués dans des collisions ou auto-collisions.

Pour remédier aux problèmes du rendu dans les cas limites (silhouette non déformée), il est possible d'utiliser du *displacement mapping* qui déplace les sommets du maillage avant le rendu. Cette technique est maintenant rapide car elle exploite les propriétés des cartes graphiques récentes. En contrepartie, il devient impossible (comme dans le cas du *bump mapping*), d'effectuer des calculs sur le maillage après cette étape car les nouvelles positions des sommets ne sont pas stockées.

Volino et al. [VMT99] proposent une technique rapide, basée sur des polygones pour animer les rides de modèles déformables. Il s'agit de moduler l'amplitude d'un schéma de rides donné sur chaque

triangle. Le maillage est automatiquement raffiné si nécessaire. Pour le rendu, il utilise soit du *bump mapping*, soit du *displacement mapping*. Ce travail qui donne de très bons résultats visuels a été étendu dans [HBVMT99]. Le problème de la technique se trouve du côté de l'utilisateur qui doit dessiner un motif de rides complet et régler des paramètres jugés non intuitifs par Kono et Genda [KG03]. Plus récemment, Bando et al. [BKN02] ont repris le même genre d'idée. La différence se situe dans la construction de la carte de déplacements. L'utilisateur doit spécifier les rides potentielles une par une dans la projection $2D$ du maillage de l'objet en dessinant une courbe de Bézier comme creux de la ride et en spécifiant quelques paramètres. Lors de l'animation, ces rides seront présentes ou pas. En plus de cette étape pénible, un pré-calcul coûteux sur la surface est nécessaire puisque le maillage ne doit pas être quelconque. Il est donc rééquilibré par une étape de minimisation d'énergie sur les arêtes de la surface polygonale. Les résultats obtenus par Bando sont présentés sur la figure 3.14.

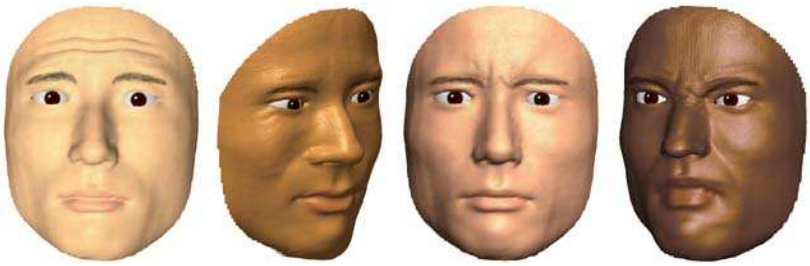


FIG. 3.14 – Résultats de rides du visage obtenus par Bando et al. en 2002 (images extraites de [BKN02]).

La modulation de l'amplitude des rides peut aussi être obtenue grâce à des modèles physiques parfois assez élaborés [BKMTK00, WTT94, WKT96, WKT97, WKMMT99]. L'idée est de coller au maximum à la réalité physique du tissu déformé et d'utiliser les paramètres d'élasticité et de viscosité de la peau pour résoudre des équations qui génèrent une position d'équilibre et des paramètres pour le rendu des rides. Dans [WKT97] par exemple, la peau est une surface triangulée qui recouvre des muscles modélisés par des B-splines. C'est donc le mouvement des muscles qui déforme le maillage et une tension est calculée en chaque triangle du maillage de la peau. Cette tension permet de calculer une élévation de la ride qui ne peut exister que sur un muscle et qui doit être alignée sur le maillage. Ces techniques sont souvent couplées à du *bump mapping* ou du *displacement mapping* pour le rendu. Elles marchent bien, mais sont incompatibles avec le temps-réel et très contraignantes. D'autre part, elles supposent que l'on connaisse les paramètres physiques de la peau ou du vêtement simulé et sont indissociables des muscles qui doivent donc être modélisés. Enfin, ce sont des méthodes peu intuitives pour un utilisateur puisque le résultat final ne peut pas être modifié directement, mais en jouant sur les paramètres d'élasticité des ressorts par exemple.

3.6.2 Déformations géométriques du maillage

Cette sous-section regroupe les articles qui modifient effectivement le maillage de l'objet (en logiciel). Cette distinction est importante puisqu'à la différence du *bump* ou du *displacement mapping*, il est possible dans ce cas d'appliquer des post-traitements (détection de collisions par exemple). Cependant, nous allons voir que beaucoup d'entre elles nécessitent d'utiliser un maillage adapté.

Rides créées à partir de splines

Les premiers travaux se sont surtout intéressés aux rides dues à l'animation du visage. Viaud et Yahia [VY92] utilisent un masque de rides de référence qui consiste en une surface spline cardinale dont les isolignes sont alignées avec les rides potentielles.

Textures de dilatation

Malgré le nom de la technique, elle ne se contente pas de calculer une texture reflétant les éventuels plis, mais génère bien un maillage $3D$ à partir d'une texture $2D$. Étant donné que les rides sont dues à une résistance à la compression, elles peuvent être générées par des techniques de simulation. Combaz et Neyret [CN02] génèrent des plis complexes en simulant les déformations statiques d'un maillage d'éléments finis, déformé par un processus de croissance interne. L'utilisateur dispose d'une interface intuitive pour peindre les directions principales des rides et leur fréquence. Ces propriétés sont utilisées pour mettre en place le comportement d'expansion anisotrope de la surface. Les résultats obtenus sont visuellement très convaincants (figure 3.15) mais la technique en elle même ne peut pas convenir pour générer des rides à surface constante en réponse à une animation, comme ce serait le cas pour animer la peau d'un personnage par exemple. Les modèles générés sont statiques.

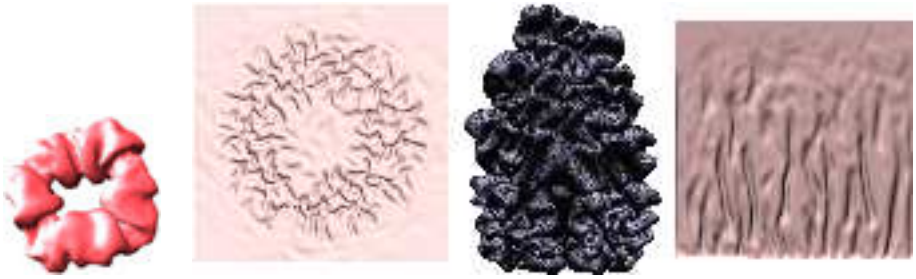


FIG. 3.15 – Résultats de plis obtenus par Combaz et Neyret en 2002 par une technique de simulation (images extraintes de [CN02]).

Préservation de contraintes

Les rides dynamiques sont dues à des contraintes de conservation de longueur inhérentes aux tissus physiques. Elles peuvent donc logiquement être générées à partir de ces contraintes. Sauvage et al. [SHB04] ont proposé un modèle de courbe multirésolution qui préserve sa longueur pendant la manipulation en ondulant à une échelle prédéfinie. Leur solution repose sur un processus d'optimisation global qui résout les contraintes sur la longueur de la courbe limite. Cette méthode, initialement loin du temps-réel, atteint maintenant des temps interactifs grâce à une approximation utilisée avant la convergence finale (résultats non publiés). Malheureusement, à l'heure actuelle, l'application de cette technique à une surface n'est pas proposée de manière élégante : les lignes à longueur constante sont forcément des lignes du maillage, ce qui donne une contrainte forte sur l'orientation des rides.

Modèles physiques

La production de films d'animation a souvent recours à des modèles physiques pour simuler les vêtements des personnages. Ces modèles utilisent les propriétés physiques des tissus pour simuler

leur comportement. Un effet de bord de ces méthodes est qu'elles provoquent la création de plis dans les endroits où le maillage se retrouve compressé afin de satisfaire un ensemble de contraintes. De nombreux travaux se sont intéressés à ce problème et un état de l'art complet de ces méthodes n'est pas le but de ce paragraphe. Néanmoins, nous citerons les travaux récents de Choi et Ko [CK02] qui produisent des plis très réalistes sur des vêtements de personnages, dans le creux du genou en particulier et le travail de Baraff et al. [BWK03] qui s'intéresse au problème des auto-collisions du vêtement et qui produit des plis très naturels.

Un inconvénient de ces méthodes est qu'elles sont loin d'être utilisables en temps-réel, dans des jeux vidéos par exemple. L'autre inconvénient est que les plis sont une *conséquence* de la simulation, et le lieu de leur apparition ainsi que leur amplitude est incontrôlable directement par l'utilisateur. Le contrôle se fait uniquement à travers des paramètres physiques tels que la raideur du matériau. Néanmoins, en contrepartie, les animations générées sont réalistes.

En bref : De nombreuses méthodes existantes pour modéliser des plis dynamiques utilisent le *bump mapping* ou le *displacement mapping*. L'idée est alors de trouver une fonction adaptée qui module l'amplitude d'un schéma de rides (une texture). Le problème de ces méthodes est qu'il n'est pas possible d'appliquer un post-traitement après le calcul de la nouvelle position des sommets du maillage car les positions calculées ne sont pas stockées. La seule possibilité dans ce cas est de coder le traitement et les post-traitements dans un *vertex program* d'une carte graphique récente. Le gros avantage de ces méthodes est qu'elles sont très rapides, les calculs lourds étant effectués par le GPU. Les autres techniques déforment le maillage $3D$, mais sont souvent très coûteuses (utilisation de simulation physique par exemple). Un autre inconvénient que l'on retrouve dans plusieurs travaux, est que les rides ne peuvent pas être placées librement, mais elles suivent le maillage et souvent, elles ne peuvent être produites que dans certaines zones prédéfinies à la modélisation. L'idéal pour respecter la contrainte de temps-réel des jeux vidéos, et simplifier le travail des artistes, serait d'avoir une technique temps-réel, intuitive avec retour visuel immédiat, et sans contraintes fortes sur la position des rides lors de la modélisation.

Conclusion

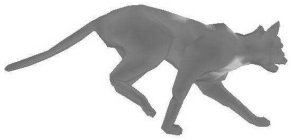
Rappelons tout d'abord l'objectif de cette thèse. Nous souhaitons ajouter des détails en temps-réel à une animation existante de personnage, comme les effets dynamiques des tissus ou bien les plis de la peau ou des vêtements. Comme nous l'avons vu au chapitre 2, le type de surface qui se prête le mieux aux applications temps-réel sont les maillages polygonaux, tout simplement pour leur rapidité d'affichage. Cependant, leur manipulation directe est impossible. Les animations sont le plus souvent effectuées grâce à l'ajout d'un squelette interne et l'utilisation d'un *skinning* pour déformer la peau. De nombreux travaux s'intéressent à l'animation du squelette lui-même afin de créer des mouvements plus naturels et plus fluides. Pour cela, l'utilisation de cinématique inverse ou de capture de mouvements permet de produire des animations de grande qualité.

Jusqu'à récemment, le problème majeur se situait au niveau du *skinning*, la technique de déformation de la surface polygonale qui utilise les repères du squelette pour calculer une forme courante de la peau. En effet, l'interpolation linéaire utilisée pour effectuer le *skinning* conduit à des écrasements et étirements du maillage au niveau des articulations. Des articles récents ont néanmoins apporté des solutions très satisfaisantes en utilisant plus de repères, plus de poids et des acquisitions de poses $3D$ réelles, tout en maintenant la qualité principale de cette méthode : le fait qu'elle soit utilisable en temps-réel.

Nous avons cependant identifié deux problèmes de la méthode. Le premier est qu'elle ne permet pas de reproduire la dynamique du mouvement. La forme créée pour une pose donnée du squelette sera la même, quel que soit le type de mouvement effectué pour arriver dans cette pose. Ce problème peut être résolu en utilisant une animation par modèle physique des tissus. Cependant, l'inconvénient des méthodes physiques est d'une part, qu'elles sont coûteuses en temps de calcul, et d'autre part, qu'elles nécessitent de passer par une représentation volumique de l'objet. Si l'on considère une animation par squelette et *skinning* existante, ces techniques ne peuvent pas lui être appliquées directement. Par ailleurs, le contrôle des déformations se fait *indirectement*, par l'emploi de paramètres physiques, ce qui est peu intuitif pour l'utilisateur et qui de plus, ne permet pas de spécifier directement le type de comportement que l'on souhaiterait obtenir.

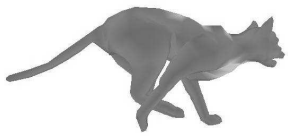
Le second besoin que nous avons dégagé au niveau du skinning est l'ajout de plis dynamiques de la peau, qui se font et se défont pendant le mouvement, à des endroits précis du personnage (creux

du poignet par exemple). Générer ces plis directement par *skinning* est impossible. Les techniques de *bump mapping* existantes donnent des résultats visuellement corrects, mais incorrects géométriquement puisque le maillage n'est pas déformé. Des techniques géométriques ont été proposées, mais elles nécessitent de *prédéfinir* au moment de la modélisation, le lieu des plis qui devront apparaître pendant l'animation. Nous ne souhaitons pas imposer ce genre de contrainte, l'infographiste qui crée les modèles et l'animateur étant la plupart du temps des personnes différentes.



Deuxième partie

Contributions



Introduction

L'objectif de cette thèse est de proposer de nouveaux outils permettant d'ajouter des détails à une animation de personnage existante afin de la rendre plus réaliste et ceci, sans altérer la rapidité d'exécution de l'animation. Les améliorations que nous allons proposer s'attaquent à deux problèmes différents, mais qui ont tous deux une incidence directe sur l'apparence externe de la peau ou des vêtements des personnages. Le premier outil, présenté au chapitre 4, a pour objectif de rajouter des déformations dynamiques dues à l'inertie de masses de chair composées de muscles et de tissus graisseux à certaines parties d'un personnage. Le second, présenté au chapitre 5, permet d'ajouter des plis dynamiques à la peau ou aux vêtements du personnage. Que ce soit l'un ou l'autre des outils, nous souhaitons satisfaire un certain nombre de contraintes, à la fois au niveau de la modularité des algorithmes et au niveau du temps de calcul de l'animation, tout en privilégiant la simplicité de l'utilisation de l'outil par un animateur.

Notre premier objectif est de créer des algorithmes exécutables en temps-réel destinés entre autres au jeu vidéo. En effet, nous avons collaboré avec la société de jeux vidéos, Galilea, dans le cadre du projet RNTL *PARI* (Personnages Animés Réalistes Interagissants). Le travail présenté dans les deux chapitres suivants est l'un des transferts de technologie envisagés pour la filiale *Virtual Actors* créée conjointement par Galilea et le laboratoire GRAVIR. Dans le contexte du jeu vidéo, les modèles couramment utilisés car plus rapides à afficher sont les maillages polygonaux. Les méthodes de simulation par modèle physique étant généralement coûteuses en calcul (même s'il est maintenant possible d'atteindre des temps interactifs), nous nous sommes plutôt dirigés vers des techniques *géométriques* pouvant, de façon très originale, exploiter quelques paramètres physiques extrêmement simplifiés.

Notre second objectif se situe au niveau de la portabilité de nos méthodes que nous souhaitons utilisables comme des couches supplémentaires sur n'importe quel modèle et/ou n'importe quelle séquence d'animation. Plutôt que de redévelopper un système complet de modélisation et d'animation tel que MuscleBuilder[3] à l'EPFL, nous avons préféré proposer différents outils, utilisables conjointement ou indépendamment, qui se greffent sur des animations classiques existantes. Ces animations de personnages peuvent être de type maillage polygonal possédant un squelette d'animation interne déformé par *skinning* (ce qui est notre cas), ou bien n'importe quelle simulation physique ou ana-

[3] http ://vrlab.epfl.ch/ aubel/MuscleBuilder/

tomique pour laquelle on souhaite ajouter des détails en temps-réel. En effet, nos outils nécessitent comme point d'entrée un maillage polygonal (ce que l'on a toujours puisqu'il faut toujours *afficher* le modèle animé à chaque pas de temps, quel que soit le type de surface employé pour sa représentation) qui est modifié à chaque pas de temps avant le rendu puis l'affichage. Si l'on se réfère à un algorithme général d'animation, nos contributions se placent au niveau du *draw()*, c'est-à-dire après l'animation et la déformation du maillage et avant le rendu de la scène à l'écran (voir algorithme 1).

Algorithme 1 Boucle d'animation et d'affichage

tant que animation en cours **faire**
  pour $i = 0$ à nb_pas **faire**
    animate() ;
  fin pour
  draw() ;
fin tant que

Il est donc possible de réutiliser n'importe quelle animation existante et d'y incorporer nos outils comme des couches supplémentaires du modèle. De plus, bien que définies lors d'une étape de modélisation, ces couches doivent avoir la possibilité d'être actives ou non lors de l'animation, ceci afin d'optimiser les temps de calcul. Par exemple, on peut imaginer que si un personnage sort du champ de la caméra ou bien se situe si loin qu'il ne prend que quelques pixels à l'écran, on désactive une couche de détails ou un élément d'une couche pour ne garder qu'une animation grossière où les détails spécifiés ne consomment pas de ressource CPU inutilement.

Enfin, notre dernière contrainte est de pouvoir fournir à l'utilisateur une spécification aisée des paramètres associés aux outils tout en lui laissant une liberté de création suffisante. Nous privilégions donc des paramètres intuitifs à régler plutôt que des paramètres ayant une réalité physique lorsque cela est possible. Par exemple, un ensemble de plis sur une portion de tissu sera spécifié par un nombre de bosses, leur espacement, leur forme, plutôt que par les propriétés physiques du tissu telles que la raideur. D'autre part, les réglages de ces paramètres sont facilités par un retour visuel en temps-réel, que ce soit avant ou pendant l'animation.

Enfin, bien que nous proposions toujours des paramètres par défaut, nous voulons que le contrôle final reste à l'utilisateur, pour lui permettre d'exprimer sa créativité. Un défaut que l'on retrouve souvent dans les modèles physiques ou anatomiques est que l'animation créée est toujours réaliste. Cependant, dans bien des cas, les personnages de films d'animation sont fictifs ou peuvent avoir besoin d'un comportement exagéré d'une réalité physique. Ainsi, même lorsque nous utilisons un modèle physique simplifié, nous laissons la possibilité à l'artiste de s'en servir pour créer des animations non réalistes.

CHAPITRE 4

Skinning dynamique

Slimy, yet satisfying !

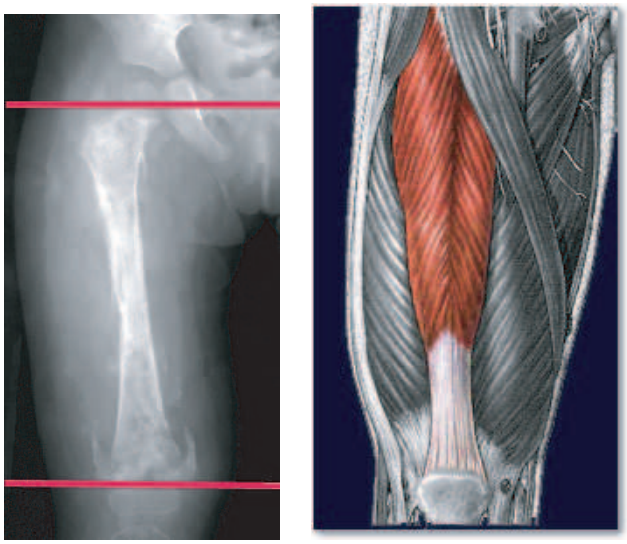
Simba, The Lion King

Ce chapitre propose un outil permettant d'ajouter des déformations dynamiques aux parties charnues d'un personnage quelle que soit la technique utilisée pour calculer les déformations statiques. En particulier, nous souhaitons pouvoir utiliser cet outil pour ajouter du réalisme à des animations déjà mises en place par des moyens classiques, comme par exemple, une animation créée à la main par un infographiste sous Maya [ali].

Les contraintes que nous nous imposons sont celles décrites dans l'introduction de la partie II, à savoir le temps-réel, la notion de couche ajoutable et la maniabilité de l'outil par l'utilisateur. D'autre part, l'effet que nous cherchons à représenter est une réponse dynamique au mouvement d'un squelette rigide interne. Afin d'obtenir des déformations réalistes (fréquence de vibration, amplitude), nous souhaitons tirer avantage des modèles physiques tout en les simplifiant au maximum pour atteindre le temps-réel.

4.1 Contraintes sur la déformation

Avant de nous lancer dans la conception d'un outil, nous allons tout d'abord, dans cette section, énumérer et justifier les contraintes que nous nous sommes fixées sur la forme, la localisation et l'induction des déformations que nous souhaitons prendre en compte.



(a) (b)

FIG. 4.1 – Une cuisse réelle : (a)-une radiographie : l'os, situé plus ou moins au centre, est entouré de muscles et de tissus ; (b)-représentation de l'enchevêtrement de muscles, tendons et tissus autour de l'os.

4.1.1 Modèle mécanique

Les déformations d'un objet quelconque soumis à des forces ne peuvent pas se calculer de manière exacte. En effet, si l'on se réfère à l'équation du mouvement d'un objet déformable présentée au chapitre 2, section 2.3.2.1, il n'existe pas de moyen de la résoudre sans discrétiser le problème en temps et en espace. Seulement, même discrétisé en éléments finis par exemple, un objet tel qu'une cuisse réelle (figure 4.1), composée de nombreux matériaux entremêlés, mènera à un système d'équations complexes que nous ne pouvons résoudre en temps-réel. C'est pourquoi il est courant de passer par une étape de modélisation de l'objet pour le simplifier en faisant des approximations sur sa forme et sur sa composition afin de calculer/intuiter les déformations connaissant les forces qui s'appliquent sur le modèle. Cette étape de modélisation amoindrit la qualité des résultats obtenus, mais réduit considérablement les temps de calcul.

Notre premier but étant le temps-réel, nous avons approché les déformations d'un membre réel par celles d'un modèle très simple. Si nous prenons l'exemple d'une cuisse (voir figure 4.1), sa composition se résume à un os rigide entouré d'un volume visco-élastique que nous supposerons homogène composé d'un mélange de fibres musculaires et de tissus adipeux. Cette approximation, bien que grossière, est suffisante étant donné que les déformations que nous cherchons à reproduire sont instantanées et que c'est surtout leur fréquence de vibration qui est perçue par l'œil humain.

Nous déduisons de ces observations un modèle théorique, très simplifié par rapport à la réalité, constitué d'un cylindre rigide qui représente l'os entouré d'un second cylindre déformable qui modélise les muscles et les tissus. Les extrémités des deux cylindres sont des disques rigides puisqu'ils se situent au niveau des articulations. Cette modélisation, bien que très simpliste, a déjà été utilisée dans les travaux de Bro-Nielsen [BN95] pour simuler des opérations chirurgicales.

La figure 4.2 décrit notre modèle et montre les déformations que nous souhaitons obtenir pour différentes accélérations. En s'appuyant sur ce modèle et sur l'observation des déformations qui ont lieu lors de mouvements réels, nous proposons trois lois de comportement de la déformation.

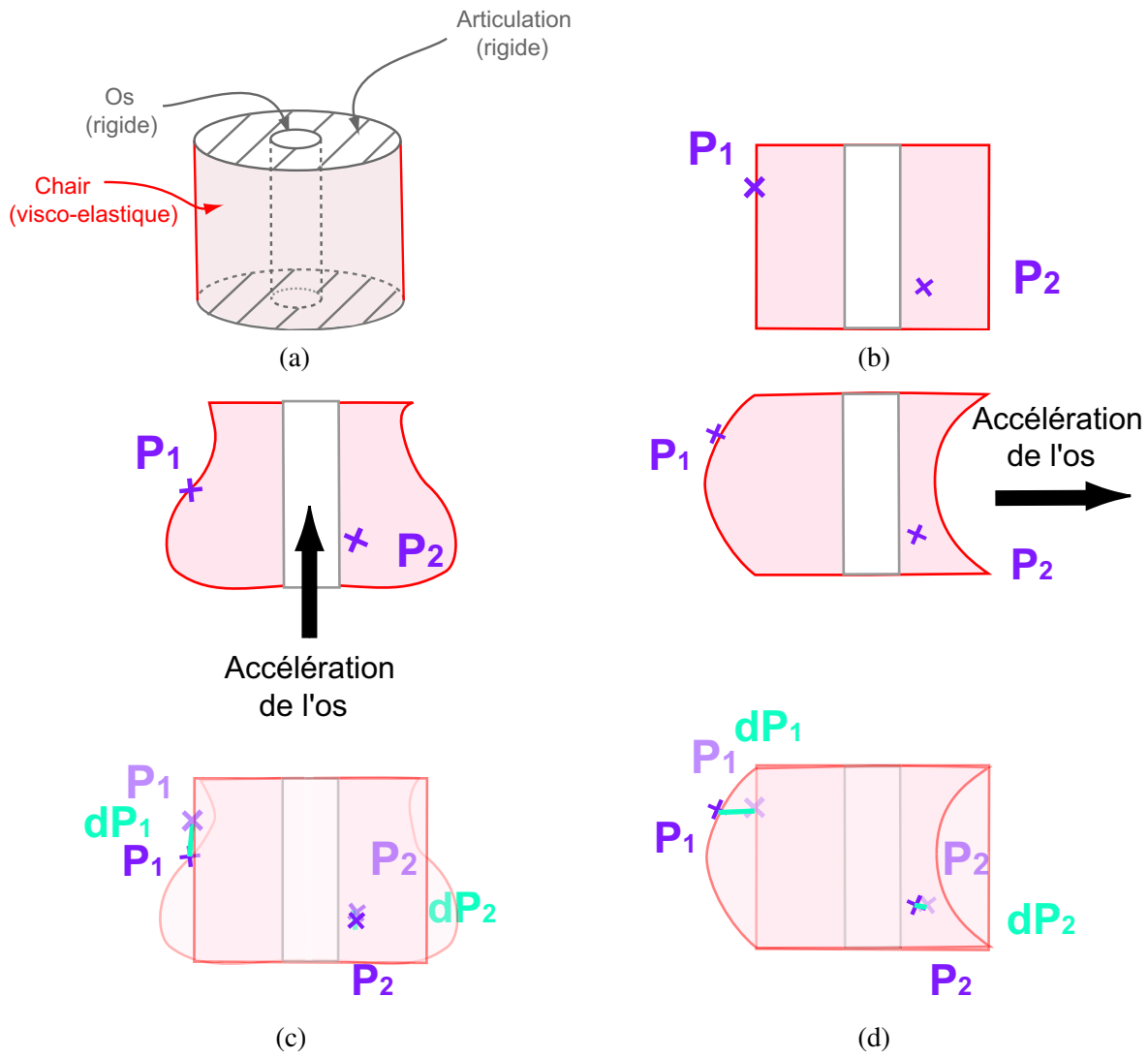


FIG. 4.2 – Un modèle mécanique simplifié d'un membre humain et ses déformations sous différentes accélérations : (a)-le modèle ; (b)-vue d'une coupe et labellisation de deux points du solide : P_1 et P_2 ; (c)-accélération de l'os dans la direction de son axe ; (d)-accélération de l'os dans la direction orthogonale à son axe ; (c) et (d)-P_1 et P_2 se trouvent déplacés dans la direction opposée à l'accélération, dP_1 et dP_2 décrivent ces déplacements.

1. Plus un point est loin de l'os, plus il est libre de se déplacer. D'autre part, le déplacement subi par un point s'effectue dans la direction opposée à l'accélération de l'os. En effet, le phénomène d'inertie entraîne les points de l'objet dans la direction de l'accélération alors que les frottements de l'air tendent à les freiner. On peut observer ce phénomène sur les points P_1 et P_2 de la figure 4.2 (b), dont les déplacements respectifs sont illustrés par les vecteurs dP_1 et dP_2 sur les figures (c) et (d).
2. Les articulations doivent pouvoir bouger sans rompre la continuité qui existe entre deux éléments consécutifs. Cela nous conduit à les modéliser comme des sections rigides qui ne subissent pas de déformations dynamiques. Nous souhaitons de ce fait que les déformations soient nulles aux extrémités de l'élément (c'est à dire aux articulations) et maximales au centre de l'élément.
3. Quel que soit le mouvement du squelette, même extrême, la chair ne doit pas pouvoir franchir l'os et se retrouver de l'autre côté. La déformation que peut subir la chair doit donc être contrainte à une déformation maximale.

La troisième contrainte peut *a priori* paraître artificielle, mais elle se justifie aisément par le fait qu'une déformation exagérée provient de forces trop importantes, d'un ordre de grandeur beaucoup trop éloigné de la réalité, qui sont appliquées au modèle. Dans la réalité, la conséquence de telles forces serait la rupture de l'os. Notre modèle actuel ne peut pas exploiter cette propriété, mais dans un travail futur, la déformation maximale autorisée pourrait être remplacée par un changement de comportement du modèle, qui ne serait plus seulement élastique mais qui pourrait prendre en compte les fractures.

4.1.2 Localisation des déformations

Les déformations dynamiques sont présentes principalement au niveau des masses musculaires et graisseuses non retenues par un os. Par exemple, un ventre, une cuisse ou bien un bras assez charnu verront leur enveloppe vibrer à une certaine fréquence lors de mouvements plutôt rapides ou violents. Ces déformations sont locales et leur fréquence et amplitude dépendent uniquement du volume de chair et de sa composition. Ces caractéristiques varient d'un membre à l'autre, d'une personne à l'autre et dépendent de certains autres facteurs tels que l'âge. D'autre part, ces effets ne sont pas présents sur la totalité du corps. Par exemple, on ne pourra pas observer ce phénomène de ballottement sur une main ou un pied, beaucoup moins charnus et plus osseux. Il est donc nécessaire d'offrir un outil qui permette de spécifier des volumes de chair indépendants ayant chacun leurs propres caractéristiques. Ainsi, ces volumes de chair peuvent être placés sur le personnage uniquement aux emplacements où ils sont nécessaires selon le type de comportement souhaité.

Le choix de l'emplacement des éléments doit aussi se faire en accord avec le respect des articulations, qui, principalement composées d'os et de peau, ne subissent pas de déformations dynamiques. On peut donc voir les articulations comme des frontières naturelles aux volumes de chair indépendants.

4.1.3 Rôle des os

Dans notre approche, comme c'est le cas dans la réalité, c'est le mouvement rigide du squelette, donc des os, qui induit une déformation dynamique des masses de chair et donc de la peau, couche externe. Les os sont donc le moteur direct des déformations et offrent une information capitale quant à la forme de la déformation et son amplitude. Si l'on se réfère à la figure 4.2 (c) et (d), on remarque que l'os *traîne* la chair et que la direction d'accélération de l'os provoque un comportement de la

chair qui est différent selon que cette accélération se fait dans la direction de l'axe de l'os ou bien dans la direction orthogonale. Pour cette raison, il sera nécessaire au sein de notre modèle d'avoir une représentation de l'os et celle-ci devra avoir une orientation proche de celle de l'os réel.

4.2 Principe général

Dans notre environnement, le modèle de personnage est constitué de deux couches : le squelette interne et la peau. La peau est un maillage triangulaire qui décrit l'enveloppe $3D$ du personnage tel que ceux présentés à la section 2.1.1. Dans notre cas, ce maillage a été créé par un infographiste, mais il pourrait tout aussi bien être issu d'une acquisition $3D$ par scanner. Les déformations statiques sont obtenues par un *smooth skinning* [ali], présenté à la section 2.3.1.2. C'est une technique complètement géométrique qui exprime la position d'un sommet du maillage déformé comme la somme pondérée de ses positions rigides dans tous les repères du squelette dont il dépend. Les poids ainsi que les dépendances aux différents repères sont spécifiés à la main par un infographiste.

Le squelette est une hiérarchie de repères qui permettent d'animer la peau. Cependant, il existe plusieurs possibilités pour placer ces repères dans le maillage. Les positions les plus fréquemment utilisées sont soit au milieu des *os*, soit aux articulations idéales (voir section 2.2.1). Ces repères ne se distinguant que par une translation, les poids du *skinning* restent les mêmes et la position de ces repères n'influe pas sur le résultat final. Dans notre cas, les repères sont placés aux articulations idéales, à la main, l'important étant que ces repères soient hiérarchiques, c'est à dire que la position d'un repère soit calculée relativement à la position de son repère père dans la hiérarchie. La racine est traditionnellement choisie au niveau du bassin. Cependant, pour les besoins de l'animation (cinématique inverse par exemple), il est fréquent de rajouter des repères qui n'ont pas de sens physique. Nous utilisons en pratique trois catégories de repères, selon leur rôle effectif :

- ceux qui servent à modéliser les articulations (un repère par articulation) et sont donc les supports du *skinning* ;
- des repères supplémentaires qui sont présents aux bouts des membres (au bout des doigts par exemple) et qui servent à la mise en place de l'animation (cinématique inverse, ajout de contraintes). Leur présence a pour conséquence qu'une portion du corps d'un personnage se retrouve toujours comprise entre deux repères au minimum ;
- des repères supplémentaires qui peuvent être rajoutés là où l'artiste le juge nécessaire pour servir de base aux calculs de notre *skinning dynamique*.

L'animation du squelette est effectuée par un infographiste, par utilisation de positions clefs, de cinématique inverse ou de capture du mouvement et nous récupérons la position de chaque repère à chaque instant pour *rejouer* le mouvement dans notre système.

L'outil que nous avons développé afin de rajouter des effets dynamiques agit localement, directement sur le maillage polygonal obtenu par *smooth skinning*. L'utilisation d'un modèle physique simplifié nous permet d'ajouter des effets physiques au sein d'un système de déformations purement géométrique. La localisation des effets recherchés nous conduit à l'introduction de la notion d'*élément de chair dynamique*. Le modèle final de personnage est obtenu en plaçant différents éléments de chair dynamique sur la géométrie. Ces éléments vibrent pendant le mouvement et permettent de modifier la géométrie de la peau obtenue par *smooth skinning* juste avant le rendu de la scène. C'est donc une méthode de *skinning* classique (ou toute autre méthode équivalente) qui gère les problèmes de raccords aux articulations et la forme statique de l'objet déformé.

L'élément de chair dynamique permet de regrouper un ensemble de sommets du maillage en une structure qui vibre autour de l'os grâce à un ressort amorti fixé au centre de celui-ci. Le ressort permet

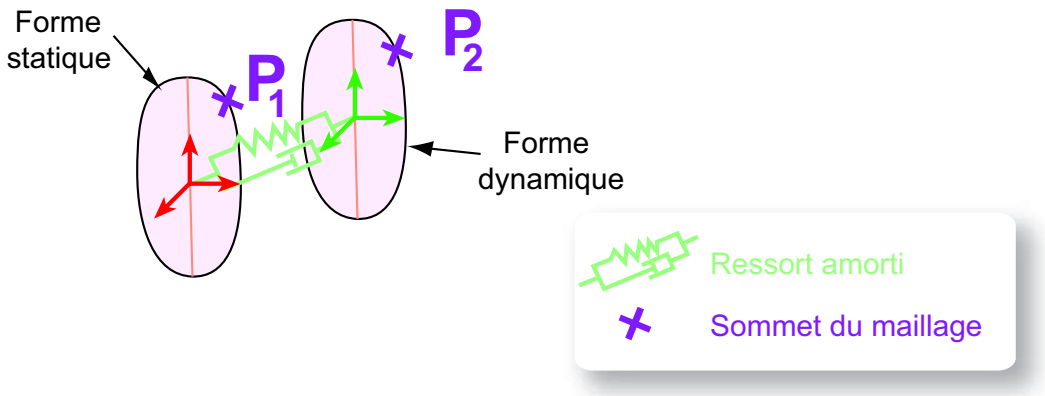


FIG. 4.3 – Deux formes clefs : la forme dynamique est la translatée de la forme statique et son mouvement est déterminé par un ressort amorti attaché à l'os.

de *traîner* un repère dynamique qui définit une forme dynamique de l'élément qui n'est rien d'autre que l'élément statique translaté. Chaque sommet du maillage de l'élément possède donc, à chaque instant, une position obtenue par *skinning* classique (point P_1 de la figure 4.3) et une position dynamique (point P_2 de la figure 4.3). La position finale de chaque sommet est obtenue par combinaison linéaire de ces deux positions. On peut voir ce processus comme une interpolation de formes clefs entre deux formes : celle obtenue par *skinning* et une autre forme, translatée de la première, qui oscille autour de l'os grâce au ressort (voir figure 4.3). Cependant, afin que la forme finale soit *déformée* et non une simple translation de la forme statique, il est nécessaire de spécifier des poids d'interpolation différents pour chaque sommet.

Lors du calcul des poids, des fonctions d'atténuation sont utilisées pour prendre en compte la localisation de la masse de chair, son volume, ainsi que la présence d'articulations, qui, essentiellement composées d'os, ne *vibrent* pas. Ainsi, une première fonction d'atténuation nous permettra de prendre en compte la morphologie du membre en favorisant le déplacement des points éloignés de l'os, alors qu'une seconde fonction nous permettra de diminuer les effets dynamiques aux abords des articulations et d'assurer la continuité entre deux éléments de chair. Ces fonctions, ainsi que le calcul des poids, seront présentés à la section 4.4, après la description qui suit de notre méthode de spécification d'un élément de chair et de son ressort associé.

4.3 Élément de chair dynamique

Cette section va nous permettre d'éclaircir la notion d'élément de chair dynamique, de préciser le rôle de l'utilisateur dans sa création ainsi que de définir son influence sur le maillage.

Appelons *élément de chair dynamique*, tout volume de muscles et de tissus dont l'enveloppe est formée d'un ensemble de sommets du maillage du modèle qui se déforment et vibrent en même temps et avec la même fréquence lorsque le repère du squelette dont ils dépendent subit un mouvement (translation ou rotation). Un élément dynamique est défini par au moins deux repères du squelette d'animation : un repère appelé *dominant* qui donne son mouvement rigide à l'élément, et un ou plusieurs repères correspondant à des fils dans la hiérarchie du squelette, qui définissent un repère *moyen* fils. Ces deux repères définissent les extrémités de l'élément et doivent être choisis sur une zone

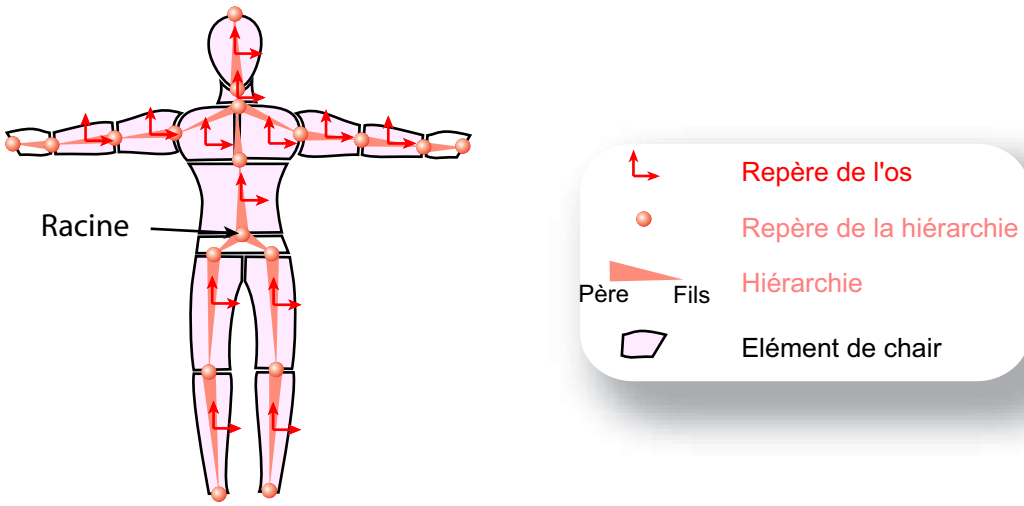


FIG. 4.4 – Un personnage articulé, un exemple de décomposition en *éléments de chair dynamique* et les systèmes de coordonnées locales correspondants.

statique du personnage qui ne subit pas de déformations dynamiques, une articulation par exemple. Ils correspondent aux deux couvercles de notre modèle mécanique (figure 4.2 (a)). Ces repères servent donc aussi de jonction entre deux éléments consécutifs, comme c'est le cas du bras et de l'avant bras, séparés par le coude. Cependant, les repères choisis ne sont pas nécessairement consécutifs dans la hiérarchie. Par exemple, pour modéliser la région du ventre, bien que le squelette d'animation puisse contenir un grand nombre de vertèbres (et donc de repères dans le squelette d'animation), un seul élément de chair encadré par deux repères éloignés suffit. L'utilisateur pourra notamment prendre le bassin comme repère dominant et une vertèbre au niveau du plexus solaire comme repère fils. Une possibilité de création d'éléments dynamiques est présentée pour un humanoïde sur la figure 4.4.

4.3.1 Spécification d'un élément dynamique

L'utilisateur spécifie un élément dynamique en sélectionnant un repère, un ou plusieurs de ses repères fils dans la hiérarchie et l'ensemble des sommets du maillage qui appartiennent à l'élément. Le repère le plus haut de la hiérarchie des repères sélectionnés sera choisi comme repère dominant (voir figure 4.5) .

Un *os virtuel* est défini par le segment de droite qui relie le repère dominant au barycentre des repères fils sélectionnés. Le milieu de l'os virtuel sera choisi comme centre du repère local de l'élément de chair et ce repère local suivra rigidement le mouvement du repère dominant au cours de l'animation. Le choix du *milieu* de l'os comme centre du repère local est motivé par le fait qu'il s'agit du *centre de gravité* de la masse de chair dans le cas idéal où le volume de chair est un cylindre centré sur l'os. Dans la réalité, les masses de chair sont généralement décentrées par rapport à l'os. Nous verrons à la section suivante comment il est possible de prendre en compte au niveau des poids du *skinning dynamique* la forme réelle du maillage tout en gardant un repère dynamique centré. Des exemples d'éléments de chair sont montrés sur les figures 4.4 et 4.6 dans différents cas simples. F_s représente le repère local de l'élément qui se situe au milieu de l'os virtuel.

Étant donné que les différents éléments se rejoignent physiquement aux articulations, c'est à dire aux repères extrémités des éléments, notre algorithme, présenté dans la suite, assure que la déformation dynamique est plus importante au centre de l'élément pour s'annuler lorsque l'on s'ap-

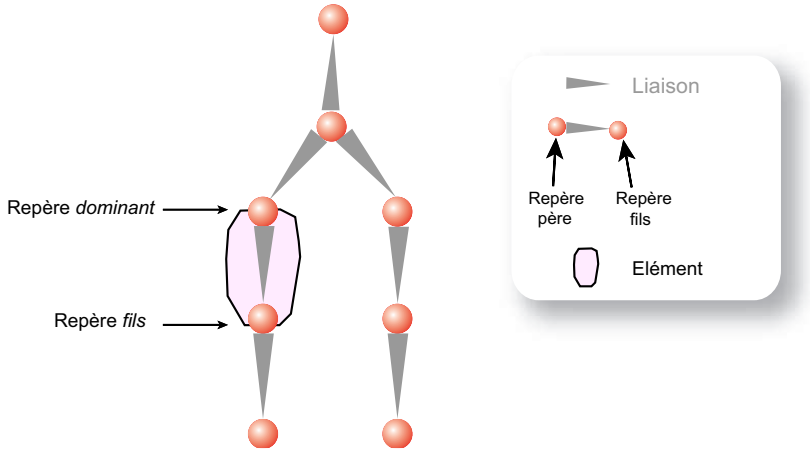


FIG. 4.5 – Un squelette d'animation et sa hiérarchie de repères contenant un élément dynamique spécifié par deux repères : le repère dominant, plus haut dans la hiérarchie, et l'un de ses repères fils.

proche de l'une des extrémités. La jonction entre deux éléments se fait ainsi sans problème et les extrémités des éléments restent indéformées, ce qui correspond à la réalité puisque les os sont des matériaux rigides qui ne se déforment pas pendant le mouvement.

4.3.2 Repère dynamique

Nous avons vu comment spécifier un élément de chair avec ses extrémités, son os virtuel, et son repère local qui se situe au milieu de l'os. Afin d'ajouter des déformations dynamiques, nous ajoutons un *repère dynamique* F_d à chaque élément de façon à capturer les vibrations visco-élastiques des muscles et des tissus lors du mouvement du squelette. Ce repère est attaché au repère local F_s par un ressort amorti de longueur à vide nulle qui se trouve être le modèle physique minimal capable de prendre en compte les propriétés visco-élastiques d'un milieu homogène (voir figure 4.6). En effet, lorsque le repère dominant est animé, il entraîne rigidement le repère F_s attaché à l'os, ce qui a pour conséquence d'*étirer* le ressort. L'élongation du ressort provoque des forces égales et opposées sur chacune des masses situées au centre de chaque repère (F_s et F_d). Le repère F_s étant contraint, ces forces n'ont aucun effet sur lui. Seul le repère F_d sera donc déplacé en réponse à ces forces (se référer au chapitre 2, section 2.3.2.2 pour plus de détails sur l'obtention des équations différentielles).

Les équations obtenues sont résolues pendant la boucle d'animation par une intégration numérique de type Euler explicite. En effet, les ressorts étant indépendants, n'importe quelle méthode simple d'intégration est suffisamment robuste pour ne pas diverger. On en déduit une nouvelle position de F_d à chaque pas de l'animation. Visuellement, cela résulte en une vibration du repère F_d autour du repère F_s car celui-ci tente de revenir à sa position d'équilibre, c'est à dire en F_s si les seules forces mises en jeu sont celles liées au ressort. Le repère dynamique, en définissant une nouvelle position de l'élément de chair, permet de faire osciller une seconde enveloppe de l'élément qui servira ensuite au mélange de formes final.

Il est important de noter que dans notre implémentation actuelle, l'orientation du repère dynamique F_d reste la même que celle du repère F_s pendant le mouvement. Ceci est une conséquence du fait que seule l'élongation du ressort est prise en compte dans la suite pour le calcul de la déformation. Cependant, il serait intéressant de se servir de ce repère dynamique pour prendre en compte une tor-

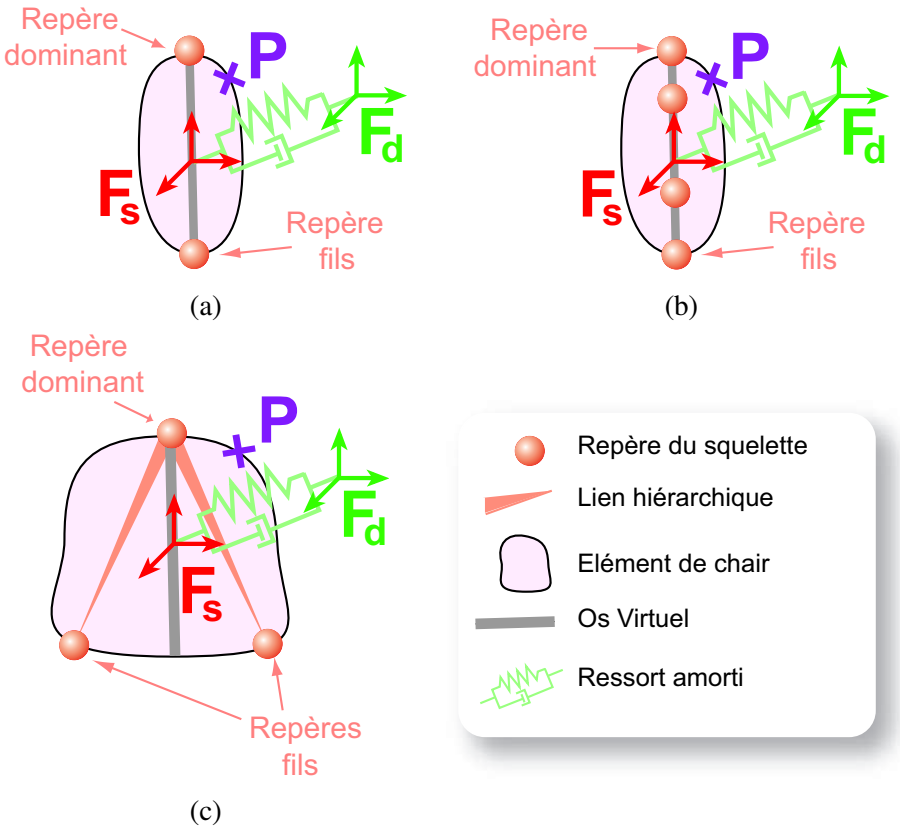


FIG. 4.6 – Un repère dynamique F_d est connecté au repère local de l'élément de chair dynamique F_s à l'aide d'un ressort amorti de longueur à vide nulle. (a) et (b) : le repère local se trouve au milieu de l'os virtuel délimité par un repère du squelette à chaque bout. (c) : le repère local est défini par 3 repères du squelette.

sion du ressort autour de son axe (voir section 4.8 pour plus de détails).

La masse ponctuelle m positionnée à l'origine du repère dynamique représente la masse de l'élément de chair. La direction et l'amplitude d'élongation du ressort modélisent l'inertie de la masse de chair. En faisant l'hypothèse d'un milieu homogène, on peut considérer que les tissus organiques sont de densité constante qui ne varie pas d'un membre à l'autre et il est donc facile de calculer une masse initiale adéquate pour l'élément à partir de son volume, tout en laissant libre choix à l'utilisateur de la modifier.

La longueur à vide du ressort est prise égale à zéro pour que pendant le mouvement, le repère dynamique tende toujours à revenir vers le repère local. Ainsi, lorsque le mouvement est nul et qu'aucune force extérieure n'agit sur l'élément, les deux repères (local et dynamique) sont confondus. Ceci se traduit par le fait que le modèle n'est pas déformé lorsqu'il est au repos.

L'amortissement a et la raideur k du ressort sont des paramètres réglables par l'utilisateur, ce qui lui permet, au cours de l'animation, de régler la fréquence des vibrations et l'amplitude des déformations. En effet, la fréquence f des vibrations est mathématiquement égale à :

$$f = \sqrt{\frac{k}{m}}$$

Une fois la raideur choisie, il faut garder à l'esprit que l'amortissement ne doit pas dépasser un

certain seuil pour ne pas être critique et pour continuer à observer des oscillations :

$$a^2 < 4km$$

Pour ne pas observer de déformations excessives, comme par exemple, la traversée de l'os d'une partie de la chair, le ressort doit être contraint à une élongation maximale. Le calcul de cette élongation maximale sera détaillé à la section 4.4.4. Lorsque le ressort atteint la valeur maximale autorisée, nous corrigeons simplement la position du repère dynamique lors de l'intégration numérique en le ramenant à sa position maximale sur la direction calculée, sans aucune modification sur la vitesse de la masse ponctuelle.

En plus des forces générées par le ressort, nous pouvons prendre en compte d'autres forces physiques comme par exemple, une pseudo-gravité. L'utilisateur peut régler la valeur de la gravité pour obtenir différentes formes au repos ou des comportements dynamiques de la chair différents. D'autres forces telles que le frottement de l'air peuvent être facilement intégrées au système.

4.3.3 Application au maillage

Les paragraphes précédents ont permis de mettre en place tous les éléments nécessaires au *skinning dynamique* (spécification des sommets concernés, ajout de repères, ajout d'un ressort amorti). Ce paragraphe nous permet de détailler le principe de la méthode et de poser le problème en équations.

À chaque pas de l'animation, les sommets du maillage du modèle sont tout d'abord déplacés par un *skinning* classique, en réponse au mouvement du squelette. Cette étape ne prend pas en compte les déformations dynamiques mais permet de calculer un maillage déformé correspondant à la pose prise par le squelette. Une seconde étape de *skinning* est ensuite appliquée sur les sommets obtenus pour calculer leur position finale. Ce second *skinning* fait une interpolation linéaire entre la position du sommet dans le repère local de l'élément de chair et sa position dans le repère dynamique. De cette façon, selon la position du repère dynamique, la chair se déforme de manière différente.

Comme dans le *skinning* classique, les deux positions du sommet contribuent de façon non égale. Il faut donc spécifier un ensemble de poids pour chaque sommet, dans chaque repère.

Soit P la position finale du sommet i. Soit P_1 la position de ce sommet dans le repère local de l'élément de chair F_s, après la première étape de *skinning*, soit P_2, la position de ce même point initial mais translaté par le ressort[1] (voir figure 4.3). Lorsque P_1 et P_2 sont exprimés dans le même repère F_s, la position finale P se calcule de la manière suivante :

$$P = \alpha_1 P_1 + \alpha_2 P_2 \tag{4.1}$$

avec α_1 et α_2, les poids de *skinning* que nous devons définir. De même que dans le *skinning* classique, nous souhaitons assurer que $\alpha_1 + \alpha_2 = 1$, pour que les déformations restent dans l'enveloppe convexe des positions définies par les deux repères.

Dans un premier temps, nous allons montrer que la déformation exprimée par l'équation 4.1 dépend directement de l'élongation $\vec{u}$ du ressort. Soit $\vec{t}$, la translation que nous devons appliquer à P_1 pour obtenir la position finale P : $\vec{t} = P - P_1$. En remplaçant P par son expression définie à l'équation (4.1), on obtient :

$$\begin{aligned}
\vec{t} &= (\alpha_1 - 1)P_1 + \alpha_2 P_2 \\
\vec{t} &= -\alpha_2 P_1 + \alpha_2 P_2 \\
\vec{t} &= \alpha_2 \overrightarrow{P_1 P_2}
\end{aligned}$$

[1] P_1 exprimé dans le repère F_s a les mêmes coordonnées que P_2 exprimé dans le repère F_d.

ce qui résulte en :

$$\vec{t} = \alpha_2 \vec{u} \tag{4.2}$$

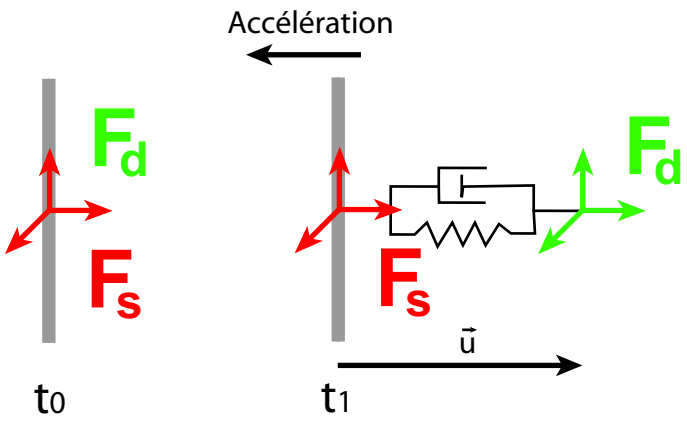


FIG. 4.7 – Comportement du ressort lors de l'accélération de l'os.

La forme finale dépend donc uniquement du poids α_2 spécifié à chaque sommet ainsi que de l'élongation et de la direction du ressort à chaque instant, exprimées par le vecteur $\vec{u}$. On remarquera que le déplacement final du point se fait donc dans la direction du ressort $\vec{u}$, qui est opposée à l'accélération (se rappeler les observations de la section 4.1.1). Le défi réside maintenant dans le calcul automatique des poids de *skinning dynamique* de façon à prendre en compte correctement les effets dynamiques simulés par le ressort amorti.

4.4 Calcul automatique des poids de *skinning dynamique*

Le ressort associé à l'élément de chair permet de lui donner une dynamique, c'est à dire une direction, une fréquence et une amplitude pour les vibrations des déformations. Cependant, ces données ne suffisent pas à prendre en compte la forme que l'on souhaite donner à la déformation. Ceci se fait à travers les poids du *skinning dynamique* qui permettent de spécifier pour chaque sommet du maillage, une capacité à se déplacer. On peut voir l'ensemble ressort + poids comme si il y avait un ressort par sommet, chacun attaché à l'os, et qui vibreraient de concert, dans la même direction et à la même fréquence, mais avec une amplitude qui dépend de l'emplacement de ce sommet. Le calcul des poids doit donc prendre en compte les trois contraintes sur la forme de la déformation présentées à la section 4.1.1.

Les deux premières contraintes sont modélisées par deux fonctions mathématiques qui sont combinées pour calculer l'ensemble des poids α_2 associés aux sommets du maillage. Afin de pouvoir exprimer simplement nos fonctions mathématiques, nous nous plaçons dans le repère local de l'élément F_s. L'origine de ce repère se situe au milieu de l'os et l'axe des ordonnées est aligné avec l'os et dirigé vers le repère dominant. La position des deux autres axes a peu d'importance. Chaque point P du maillage a des coordonnées (x, y, z) dans ce repère. La première fonction associe à P un réel qui ne dépend que de ses coordonnées x, z. La seconde fonction lui associe un coefficient compris entre 0 et 1 qui ne dépend que de la coordonnée y du point.

4.4.1 Morphologie de l'élément

La première contrainte, qui stipule que les sommets les plus loin de l'os sont les plus libres de bouger, est directement liée à la morphologie du membre, c'est à dire, à l'épaisseur de muscles et de tissus graisseux autour de l'os. Si l'on se réfère à l'équation (4.2), on remarque que plus un sommet a un poids élevé, plus il se déplace. Le poids α_2 est donc directement lié à l'épaisseur de chair, c'est-à-dire à la distance du sommet à l'os. Appelons *shape(x,z)* cette fonction :

$$shape(x, z) = d(P, B) \tag{4.3}$$

avec P, la position du sommet dans le repère local après *skinning* classique, B, le segment de droite représentant l'os virtuel, et d, la distance Euclidienne du sommet du maillage à l'os (voir figure 4.8).

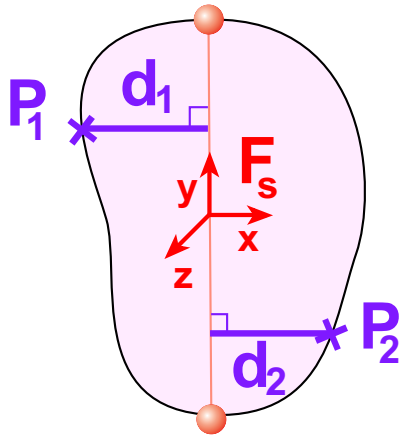


FIG. 4.8 – Le poids avant atténuation donné par la première contrainte correspond à la distance Euclidienne entre le sommet considéré et l'os virtuel de l'élément. Par exemple, P_1 aura comme poids d_1 et P_2, d_2.

Cette fonction a la bonne propriété d'être linéaire en plus d'être simple à calculer. Cependant, elle ne traduit pas forcément la réalité qui pourrait très bien mettre en jeu une fonction quadratique. Malheureusement, pour pouvoir choisir une élongation maximale pour le ressort qui garantisse à elle seule que la chair ne traversera pas l'os, l'équation *shape(x,z)*, comme on le verra plus loin, doit être linéaire. Ce choix nous a donc paru le plus judicieux, d'autant plus que nous n'avons à notre disposition aucune donnée sur la déformation réelle. Il serait cependant possible de mesurer cette déformation à partir d'une simulation non temps-réel par éléments finis par exemple.

Afin d'accentuer les poids forts et de diminuer ou de forcer à zéro des poids plus faibles, il est possible de prendre en compte une épaisseur de l'os, soit déduite automatiquement du maillage, soit réglée à la main par l'utilisateur. La façon la plus simple de procéder est de transformer le segment de l'os virtuel en un cylindre de rayon $epaisseur$. Cependant, dans le cas où le modèle possède une description géométrique du squelette, il est plus juste et plus intéressant de prendre en compte l'épaisseur réelle de l'os au niveau du point concerné. Dans les deux cas, il suffit de soustraire un scalaire à la distance calculée par l'équation (4.3) et qui correspond à l'épaisseur au point P. L'équation $shape(x, z)$ devient simplement :

$$shape(x, z) = d(P, B) - epaisseur(P) \tag{4.4}$$

Ainsi, l'équation (4.4) est utilisée lorsque l'on souhaite réellement modéliser les os avec une surface géométrique, ou bien si l'on veut limiter la capacité de compression de la chair qui se trouve

autour. Ces fonctions nous donnent donc une première approximation du poids final qui tient compte de la morphologie du membre simulé. Ce poids est ensuite atténué aux extrémités avant d'être normalisé.

4.4.2 Atténuation vers les extrémités

En effet, les articulations étant rigides, nous souhaitons que les déformations soient nulles aux extrémités des membres et, donc, dans le cas classique où un élément dynamique est associé à un membre, aux extrémités de l'élément. En accord avec cette contrainte, nous souhaitons en toute logique que les déformations soient maximales au centre de l'élément. D'autre part, les éléments se rejoignant éventuellement aux extrémités, maintenir une déformation dynamique nulle à chaque bout permet d'éviter d'éventuels problèmes d'auto-intersection ou d'étirements trop importants du maillage et d'éviter ainsi les problèmes de continuité entre deux éléments juxtaposés. Nous appliquons donc une atténuation des poids obtenus par $shape(x, z)$ par une valeur variant continûment de 0 aux extrémités de l'élément (pas de déformation) à 1 au centre de l'élément (déformation non atténuée).

Soit *attenuation(y)*, la fonction mathématique qui cherche à satisfaire cette contrainte. Elle pénalise le déplacement d'un point lorsqu'il se rapproche des repères définissant l'élément de chair. Rappelons que notre repère local F_s a été choisi pour se situer au centre de l'os. D'autre part, étant donné que nous connaissons la longueur de l'os, il est facile de normaliser les coordonnées y de nos points du maillage afin qu'elles varient de 1 (extrémité dominante) à -1 (l'autre extrémité). Nos contraintes sur la fonction recherchée peuvent donc s'écrire de la manière suivante (voir figure 4.9) :

- $attenuation(y) = 0$ pour $y = -1$,
- $attenuation(y) = 1$ pour $y = 0$,
- $attenuation(y) = 0$ pour $y = 1$.

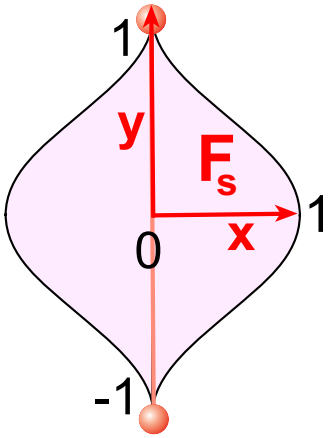


FIG. 4.9 – La fonction d'atténuation diminue les poids obtenus par $shape(x, z)$ d'autant plus que sommet concerné est proche d'une des extrémités de l'élément.

La fonction mathématique continue la plus simple qui satisfasse ces contraintes est une parabole :

$$attenuation(y) = 1 - y^2 \tag{4.5}$$

Cependant, cette fonction ne décroît pas avec une tangente nulle en -1 et 1, ce qui se traduit par des discontinuités au niveau de la jonction entre deux éléments de chair ou bien entre un élément de

chair et de la géométrie non dynamique. Ce problème peut être résolu par l'utilisation de fonctions en cloche comme la fonction polynomiale de Wyvill [WMW86], moins coûteuse en temps de calcul qu'une gaussienne classique :

$$attenuation(y) = 1 + \frac{-4 * y^6 + 17 * y^4 - 22 * y}{9} \tag{4.6}$$

Bien sûr, n'importe quelle autre fonction satisfaisant ces contraintes peut être utilisée, comme par exemple, les splines de Hermite, qui pourraient permettre à l'utilisateur de jouer avec les tangentes et les amplitudes pour obtenir une déformation de forme plus originale. En effet, rien n'oblige notre fonction à être symétrique. Les seules contraintes sont qu'elle doit être continue au minimum C1 pour observer une déformation lisse, qu'elle doit s'annuler à chaque extrémité, et de préférence, avec des tangentes nulles. Il faut aussi garder à l'esprit que cette fonction d'atténuation donne la forme globale de la déformation, un peu comme si notre élément de chair était un élément fini.

La grande variété des fonctions possibles ne fait que compliquer notre choix. Par exemple, il serait tout à fait possible de laisser le soin à l'utilisateur de *dessiner* cette fonction à l'aide d'une interface adaptée. Cependant, cela requiert des talents d'artiste que n'ont pas forcément tous les utilisateurs et que nous n'avons pas en particulier. C'est pourquoi nous avons préféré orienter notre choix vers des fonctions facilement implémentables et symétriques telles que la fonction de Wyvill ou la parabole. En effet, bien que cette dernière ne décroisse pas avec des tangentes nulles aux extrémités, elle produit des résultats plus proches d'une déformation réelle que la fonction de Wyvill. Dans notre implémentation, nous proposons donc la parabole et la fonction de Wyvill pour ses qualités de décroissance et sa rapidité de calcul. Par ailleurs, nous verrons que ce sont les mêmes fonctions qui sont utilisées pour la modélisation de plis, présentée au chapitre suivant. La figure 4.10 montre différentes déformations obtenues dans les mêmes conditions sur un cylindre idéal, mais avec des fonctions d'atténuation différentes.

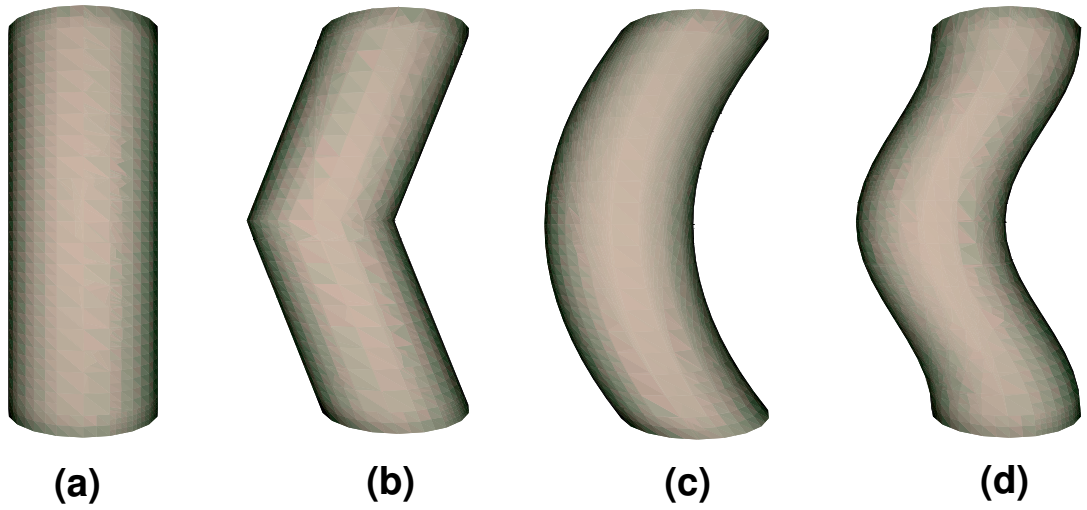


FIG. 4.10 – Résultat de l'utilisation de différentes fonctions d'atténuation sur un élément de chair cylindrique ayant une épaisseur nulle de l'os : (a)-le cylindre non déformé ; (b)-l'utilisation d'un schéma linéaire non réaliste nous montre la nécessité de l'utilisation d'une fonction au minimum C1 ; (c)-la fonction parabole offre une déformation réaliste mais engendre des problèmes de continuité aux extrémités ; (d)-la fonction polynomiale de Wyvill décroît avec une tangente nulle aux extrémités pour des temps de calcul raisonnables.

4.4.3 Normalisation

Le poids final α_2 est une combinaison de *shape(x,z)* et de *attenuation(y)*. Les deux fonctions sont multipliées puis le poids obtenu est normalisé de telle sorte que les poids varient toujours de 0 à 1 :

$$\alpha_2(x, y, z) = \frac{shape(x, z) * attenuation(y)}{max(shape * attenuation)} \tag{4.7}$$

avec $max(shape * attenuation)$, la valeur maximale obtenue parmi tous les sommets de l'élément de chair. Cette normalisation est nécessaire puisque des poids basés uniquement sur la distance à l'os, même avec une atténuation, dépendraient directement de l'échelle du modèle. Ceci résulterait en des déformations ou trop petites, ou complètement irréalistes.

Un exemple de courbe des poids obtenus est montré sur la figure 4.11 pour un élément de chair irrégulier dont le profil est tracé en (a) et deux fonctions d'atténuation différentes présentées en (b). Le résultat de la combinaison des deux fonctions est montré en (c).

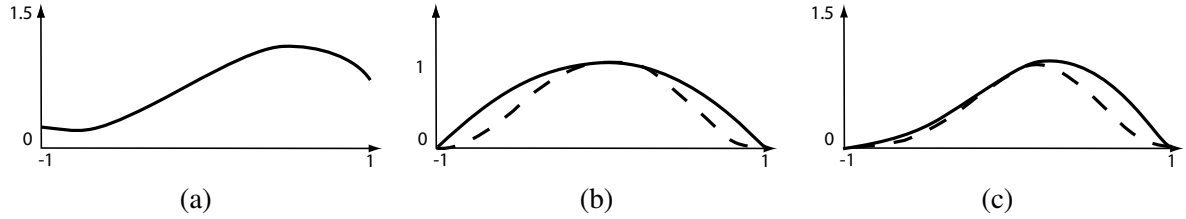


FIG. 4.11 – Effets de la fonction d'atténuation : (a)-un exemple de shape(x) ; (b)-la fonction d'atténuation en forme de parabole (trait plein) et la fonction d'atténuation de Wyvill (en pointillés) ; (c)-la fonction des poids correspondante avant normalisation, pour chaque fonction d'atténuation présentée en (b). Observez la tangente non nulle présente sur la courbe des poids obtenue avec la parabole comme fonction d'atténuation (trait plein), alors que la courbe obtenue grâce à la fonction de Wyvill a des tangentes nulles, ce qui évite les discontinuités du maillage après l'étape de *skinning dynamique* (en pointillés).

Enfin, nous tenons à attirer l'attention sur le fait que les poids α_2 ne sont calculés qu'une seule fois, pendant l'initialisation et que ce calcul est immédiat. Cependant, l'utilisateur peut, au cours de l'animation, changer l'épaisseur de l'os ou la fonction $attenuation$ auquel cas les poids sont recalculés.

4.4.4 Élongation maximale du ressort

Quel que soit le mouvement du squelette, même extrême, la chair ne doit pas pouvoir franchir l'os et se retrouver de l'autre côté. La déformation que peut subir la chair doit donc être contrainte à une déformation maximale. Par défaut, nous considérons que le seuil critique de déformation est atteint quand la chair a une épaisseur nulle à une position donnée autour de l'os. Ceci revient à spécifier, pour chaque point, un déplacement maximal à atteindre. Or, comme le montre l'équation 4.2, la position finale d'un sommet P est donnée par $\alpha_2(P) * \vec{u}$, c'est à dire par une combinaison de l'élongation du ressort, qui est la même pour chaque point, et le poids de ce point, choisi individuellement pour refléter la capacité de ce point à se déplacer. Ainsi, le moyen évident d'imposer un déplacement maximal à tous les points est d'imposer une élongation maximale au ressort.

Cependant, pour que cette élongation maximale ait un sens, elle doit être calculée automatiquement à partir des poids générés. En effet, si l'on choisit, avant la normalisation, le poids α_2 le

plus élevé de tous les points comme élongation maximale, on peut garantir qu'aucun des points ne se déplacera de plus de son épaisseur. En effet, le sommet ayant le poids maximal est le seul qui bénéficie d'un déplacement complet, égal à la valeur du ressort. En choisissant le poids avant normalisation à ce point comme élongation maximale on sait que la déformation maximale correspondra à la différence entre la courbe représentant le profil de l'élément et la courbe des poids. Ainsi, il suffit que la courbe des poids avant normalisation ne soit jamais au dessus de celle du profil. C'est pourquoi la fonction $shape(x, z)$ choisie doit être linéaire. En prenant l'équation que nous avons choisi, la courbe $shape(x, z)$ est la même que celle du profil de l'élément. Ainsi, quelle que soit la courbe d'atténuation choisie, la courbe des poids avant normalisation se situera toujours au dessous de celle du profil.

La figure 4.12 (a) illustre ceci en superposant le profil de l'élément de chair, en trait plein, et les courbes de poids obtenues pour différentes atténuations. La courbe des poids avant normalisation obtenue par l'atténuation parabolique est représentée en pointillés fins, celle obtenue par la fonction de Wyvill en pointillés larges. Sur la figure 4.12 (b) par contre, on peut voir ce qu'il se passe si l'on choisit une courbe non linéaire pour shape, par exemple $shape(x, z) = d(P, B)^2$. La courbe des poids obtenue par atténuation parabolique *croise* la courbe du profil de l'élément : l'élongation maximale ne peut plus garantir à elle seule que la chair ne traversera pas l'os.

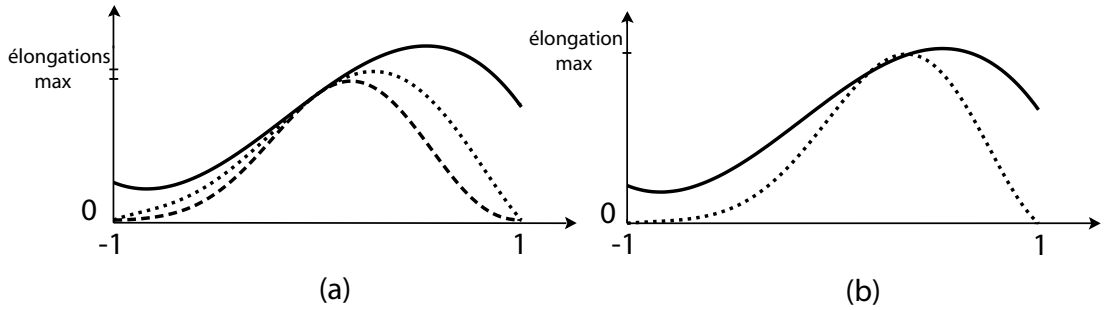


FIG. 4.12 – Un exemple de fonction $shape$ non linéaire qui ne garantit pas que la chair ne traversera pas l'os : (a)-la fonction $shape$ (en trait plein) est linéaire et les courbes de poids avant normalisation (atténuation parabolique en petits pointillés, atténuation de Wyvill en pointillés larges) ne croisent jamais la courbe $shape$. L'élongation maximale correspondante peut garantir que la chair ne traversera jamais l'os ; (b)-la fonction $shape$ non linéaire pose des problèmes avec l'une des courbes d'atténuation (parabole) qui la *croise*. L'élongation maximale ne peut pas garantir à elle seule que la chair ne traversera pas l'os.

L'élongation maximale imposée au ressort est prise en compte dans la boucle d'animation, lors de l'étape d'intégration numérique qui consiste à calculer la nouvelle position des repères dynamiques. Elle permet ainsi, quelle que soit l'amplitude des mouvements, d'empêcher qu'un sommet du maillage ne *traverse* le segment représentant l'os virtuel. Si l'on veut accorder une épaisseur non nulle comme épaisseur limite à la déformation, il est possible d'agir à deux niveaux : soit en donnant une épaisseur à l'os lors du calcul des poids, soit en diminuant la valeur accordée à l'élongation maximale.

4.5 Limitations et solutions

Lors de la spécification des éléments de chair dynamique, des problèmes peuvent apparaître dans les cas où plus de deux repères du squelette sont utilisés pour définir un élément et lorsque celui-ci est

décentré. En effet, notre algorithme est basé sur le modèle d'élément présenté sur la figure 4.2, modèle qui reste valide pour des éléments de chair plus ou moins cylindriques. Ceci est donc bien adapté pour les membres tels que le bras, l'avant-bras, la jambe ou la cuisse, mais peut l'être beaucoup moins dans la région des épaules par exemple. L'utilisateur doit donc être vigilant lors de la spécification des éléments, et il peut être souhaitable de rajouter un repère dans le squelette d'animation afin de définir des éléments plus ou moins cylindriques. Un exemple d'élément dégénéré est montré sur la figure 4.13 (a), où l'os est décentré par rapport au volume de chair. Une solution possible est de rajouter un repère (fig 4.13 (b)) pour se ramener au cas cylindrique. Cette figure est à comparer à la figure 4.6 (c), où, pour un même nombre de repères utilisés, tout se passe bien.

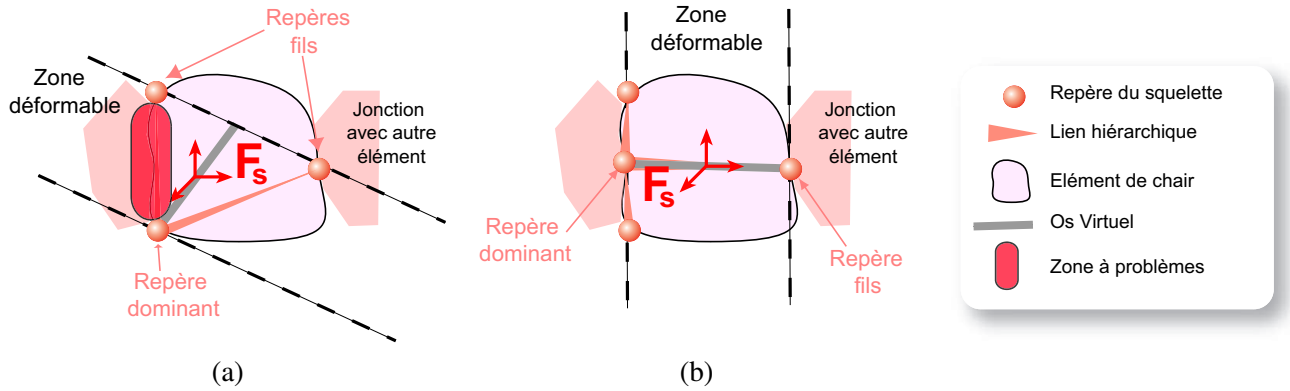


FIG. 4.13 – Un exemple d'élément dégénéré : (a)-l'élément dégénéré avec mise en évidence de la zone à problèmes ; (b)-ajout d'un repère au squelette pour résoudre le problème.

4.6 Algorithme et performance

Notre algorithme peut se décomposer en deux parties, car il provoque des calculs à deux niveaux différents de la boucle d'animation : au niveau de la fonction *animate()* pour le calcul des déformations des ressorts et au niveau du *draw()* pour la prise en compte effective du *skinning dynamique* (se référer à l'algorithme 1 pour resituer ces deux fonctions).

La fonction *animate()* (voir algorithme 2) se charge de mettre à jour les positions et orientations des repères F_s et F_d. La position/orientation des repères F_s est déterminée par le mouvement du squelette d'animation (première boucle *pour*). Ensuite, il faut faire l'intégration numérique des ressorts qui doit s'assurer que la nouvelle position de F_d n'induit pas une élongation du ressort supérieure à $elongation_{max}$. Cette étape d'intégration est la plus coûteuse de l'algorithme puisqu'elle induit une série de multiplications/divisions. Mais son coût reste très faible. D'autre part, les ressorts n'étant pas en interaction, la complexité de la fonction *animate()* est linéaire en nombre de ressorts.

Ensuite, au niveau du *draw()* (voir algorithme 3), après une étape de *skinning* classique, il faut effectuer le *skinning dynamique* qui consiste à calculer la nouvelle position des sommets du maillage en fonction de leur poids et de l'élongation du ressort. L'opération est simple et peu coûteuse puisqu'il s'agit, pour chaque sommet du maillage, d'une simple multiplication suivie d'une addition. L'algorithme est donc aussi linéaire en nombre de sommets du maillage.

Algorithme 2 fonction animate()

```
(...)
MAJ des repères locaux ;
/* intégration des ressorts */
RAZ forces
si gravité alors
    forces += m * g ;
finsi
/* ajout des autres forces */
pour chaque ressort faire
    direction = position courante repère local - position pas précédent repère dynamique ;
    élongation = direction.norme() ;
    si élongation > 0 alors
        direction.normalisation() ;
        force = forces[ressort] + élongation*direction*raideur - vitesse repère dynamique ;
        accélération = force / masse ;
        vitesse += accélération * dt ;
        position += vitesse * dt ;
        /* Correction si l'élongation max est dépassée */
        si nouvelle longueur ressort > elongation_max alors
            position = position repère local + direction nouvelle élongation * elongation_max ;
        finsi
    finsi
fin pour
```

Algorithme 3 fonction draw()

```
(...)
/* Calcul de la position des sommets par smooth skinning */
si éléments dynamiques alors
    pour chaque elt e faire
        élongation ressort = position repère local - position repère dynamique ;
        si élongation > 0 alors
            pour chaque point p de e faire
                position += poids * élongation ;
            fin pour
        finsi
    fin pour
finsi
(...)
MAJ normales ;
/* Affichage des triangles */
```

4.7 Quelques Résultats

Avant toute chose, il convient de préciser que les résultats d'un algorithme ajoutant des effets dynamiques à une animation sont très difficiles à apprécier à travers des images fixes. Il serait plus approprié de se référer aux vidéos disponibles à l'adresse suivante :
http ://www-evasion.imag.fr/Publications/2004/Lar04/

Néanmoins, l'effet de notre outil est représentable par le champ de déplacement des sommets, entre leur position issue du *skinning* classique et leur position issue du *skinning dynamique*. Nous avons représenté ce champ par des petits traits (bleus) qui montrent à la fois que notre outil agit bien, et que cette action légère et subtile est conforme aux contraintes que nous nous étions fixées à la section 4.1.

4.7.1 Un cas simple : le champignon

Notre technique a été testée avec succès sur un modèle très simple de champignon. Avant d'animer ce modèle, nous avons placé un squelette constitué de deux repères uniquement au centre du modèle. Le repère dominant est placé au niveau du chapeau du champignon alors que son repère fils se situe à la base du pied. Nous avons créé un seul élément de chair, englobant tout le maillage et dépendant des deux repères du squelette. Le ressort responsable des mouvements dynamiques est donc situé au milieu du pied du champignon et provoque des vibrations lorsque le modèle est en mouvement (voir figure 4.14).

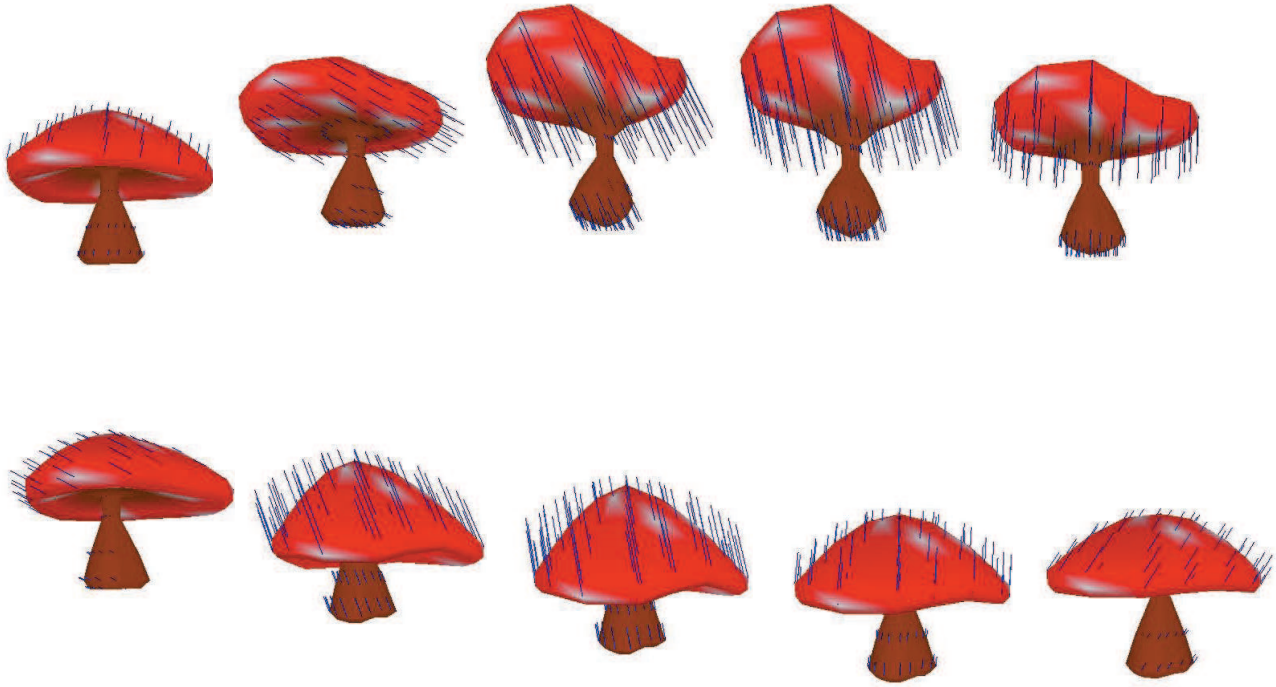


FIG. 4.14 – Un modèle de champignon surfacique déformé par notre méthode de *skinning dynamique*. Le mouvement est induit par un squelette rigide placé au milieu du modèle et qui est défini par deux repères : le repère dominant au niveau du chapeau du champignon et son repère fils à la base du pied. Les déformations obtenues sont à comparer à celles de la figure 7.1 qui a été simulée par un réseau volumique de masses et de ressorts. Avec notre technique, un seul ressort amorti a été nécessaire.

Les déformations obtenues par ce modèle simple sont à comparer avec celles que l'on obtient par une simulation physique $3D$, présentées sur le même modèle à la figure 7.1.

D'autre part, il est possible de simuler la présence de forces supplémentaires en dehors de celles générées par le mouvement de l'os. Ainsi, nous avons simulé notre champignon au repos, soumis à différentes gravités. Sur la figure 4.15, on peut voir un champignon de masse totale $39.35\ Kg$, soumis à une gravité de $16.3\ N.Kg^{-1}$ en (a), de $36.4\ N.Kg^{-1}$ en (b) et de $63.4\ N.Kg^{-1}$ en (c).

(a) : $g = 16.3\ N.Kg^{-1}$ (b) : $g = 36.4\ N.Kg^{-1}$ (c) : $g = 63.4\ N.Kg^{-1}$

FIG. 4.15 – Notre modèle de champignon de masse $39.35\ Kg$, soumis à une gravité variable : plus la gravité est forte, plus le modèle est déformé.

4.7.2 Exemple de mouvement rapide sur un animal

Notre second modèle est le chat que nous avons déjà rencontré au chapitre 2. Nous avons souhaité reproduire les ballottements du ventre de l'animal lorsque celui-ci court. Pour cela, bien que le dos du chat soit constitué d'un grand nombre de vertèbres, un seul élément de chair a été utilisé. Il englobe le maillage compris entre la première vertèbre après le bassin et la dernière vertèbre avant le cou. Le ressort se retrouve donc attaché approximativement au milieu du dos de l'animal (voir figure 4.16).

Une deuxième série d'images décrit des déformations similiaires obtenues sur le même modèle mais dans le cadre d'un mouvement complètement différent. Sur la figure 4.17, le chat effectue un mouvement de saut, constitué de forts changements d'accélération du squelette.

4.7.3 Animation d'un humanoïde

Notre exemple suivant montre l'animation d'un humanoïde. Sur les figures 4.18, 4.19 et 4.20, nous montrons les poids et les déformations générées par un seul élément de chair englobant la poitrine.

Sur notre second exemple, nous avons créé quatre éléments de chair : un au niveau de la poitrine, deux au niveau des bras et un associé à la région du ventre et des fesses du personnage. La première rangée de la figure 4.21 montre les positions des différents ressorts amortis ainsi que les poids associés alors que la seconde rangée montre quelques images tirées de l'animation. Cet exemple permet d'illustrer le fait que plusieurs éléments de chair peuvent être combinés et que les fonctions d'atténuation utilisées sur chaque élément de chair assurent que l'animation générée ne comporte pas de discontinuité au niveau du maillage.

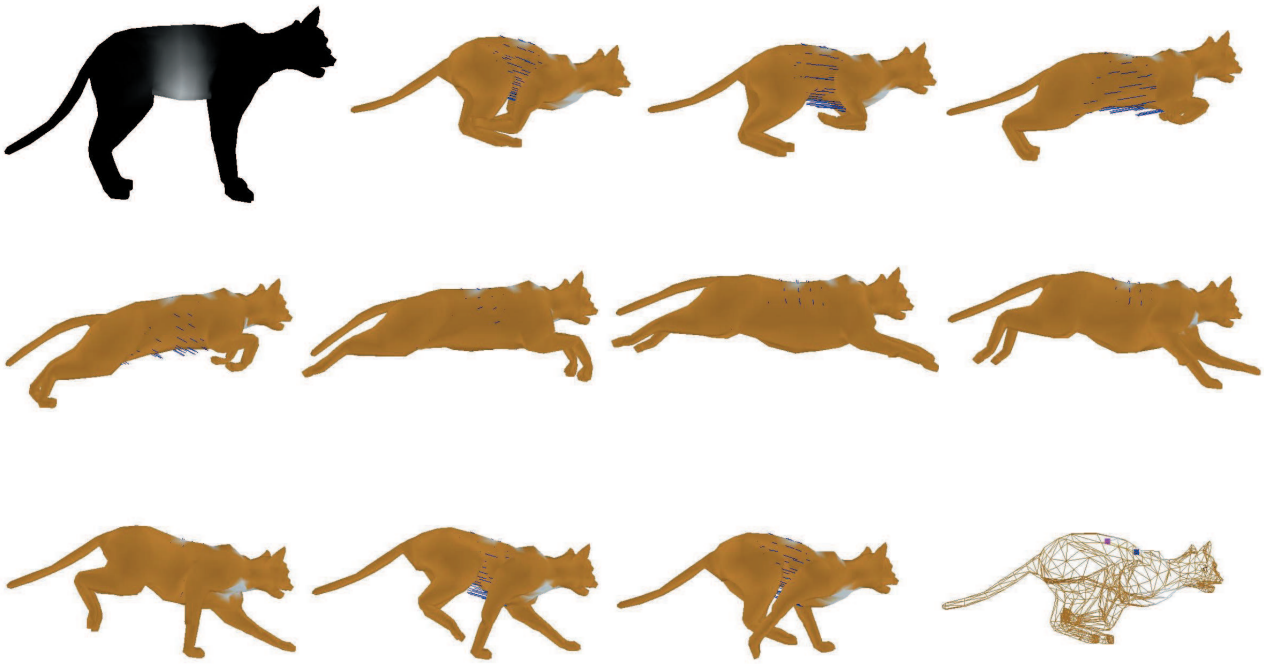


FIG. 4.16 – Animation d'un modèle de chat, en temps-réel, en utilisant notre méthode de *skinning dynamique*. La première image montre les poids associés aux sommets du maillage (ils varient de 0 en noir à 1 en blanc). La dernière image montre le ressort (le carré rose est le point fixe, le bleu est l'extrémité libre du ressort).

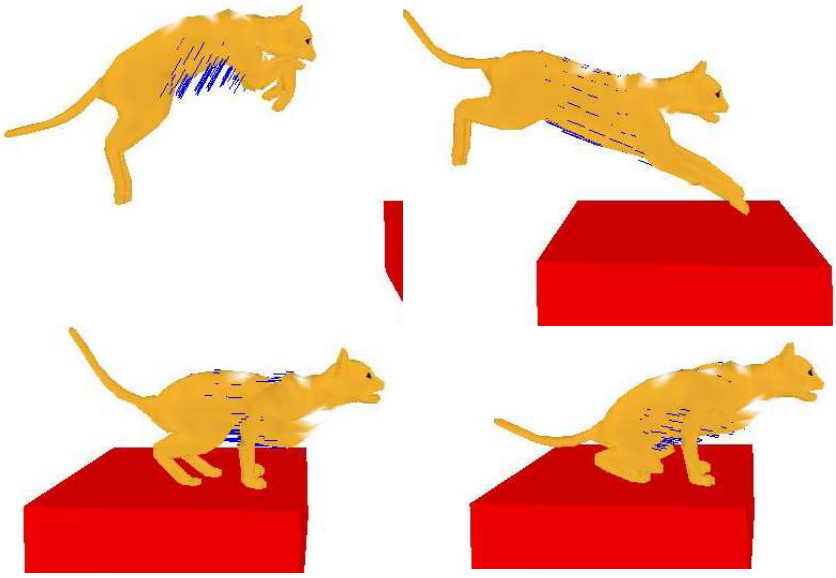


FIG. 4.17 – Le même modèle de chat, déformé en utilisant le même élément de chair dans la région du ventre, dans un mouvement de saut.

FIG. 4.18 – Animation de la poitrine d'un personnage : l'image de gauche montre l'ensemble des valeurs des poids (variant de noir pour 0 à blanc pour 1) ; l'image de droite montre le modèle non déformé dans sa position de repos.

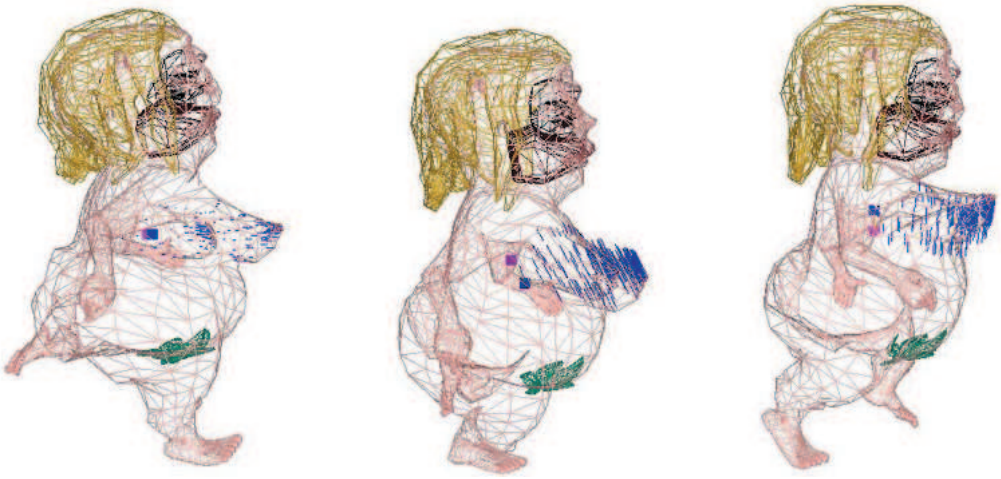


FIG. 4.19 – Animation de la poitrine d'un personnage : visualisation des déformations en affichage *fil de fer* ainsi que du mouvement du ressort amorti.

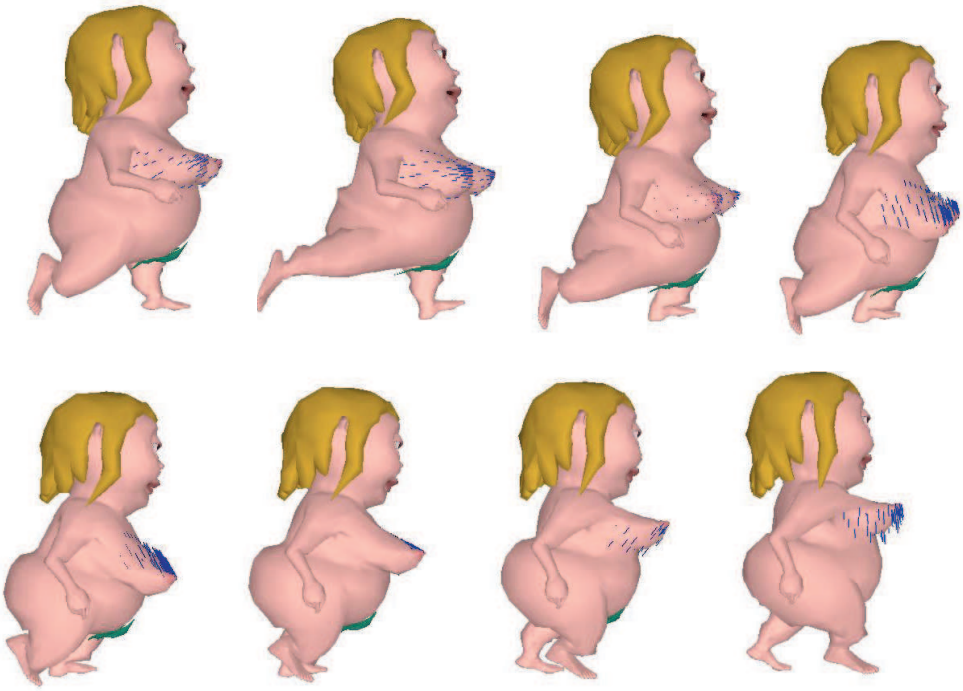


FIG. 4.20 – Animation de la poitrine d'Eve par un élément de chair dynamique.

4.8 Conclusion et perspectives

Nous avons proposé, dans ce chapitre, une technique temps-réel permettant d'ajouter des déformations dynamiques aux parties charnues d'un personnage dont les déformations cinématiques ont été calculées par *skinning*. Notre *skinning dynamique* utilise un modèle physique très simple qui, associé à un ensemble de poids, permet de calculer des déformations complexes. L'utilisateur peut contrôler les déformations à plusieurs niveaux grâce à des paramètres intuitifs : directement au niveau du ressort en jouant avec les paramètres élastiques de ce dernier ainsi que sur la masse de la zone de chair représentée par l'élément dynamique, et au niveau du calcul des poids de skinning, par l'intermédiaire de fonctions. Le gain important en temps de calculs sera montré au chapitre 7 où l'on compare l'animation d'un même modèle par deux techniques différentes : une simulation physique par masses-ressorts et notre *skinning dynamique*.

Ce travail, bien que donnant des résultats satisfaisants, ouvre la voie à de nombreuses perspectives. La première question qui vient à l'esprit en ce qui concerne le ressort est celle de la rotation du ressort. Dans l'implémentation actuelle, son orientation n'est pas prise en compte. Il serait pourtant possible de prendre en compte une certaine *torsion* du ressort, et de tirer pleinement profit du fait que les repères F_s et F_d sont des repères $3D$ qui pourraient très bien avoir leur orientation propre. Afin d'orienter relativement ces repères au cours du mouvement, il est possible de rajouter un couple de torsion comme ce que l'on peut retrouver dans les travaux de McNeely et al. [MPT99] sur le retour d'effort. Le modèle serait alors celui présenté à la figure 4.22 (a). Cependant, il est possible que le calcul des angles et des vitesses angulaires soit trop coûteux. Cette torsion pourrait alors être induite à l'aide de trois ressorts au lieu d'un, chacun reliant les axes des repères F_s et F_d deux à deux (voir figure 4.22 (b)). Notre algorithme de ce fait serait environ trois fois plus coûteux (un peu plus car il

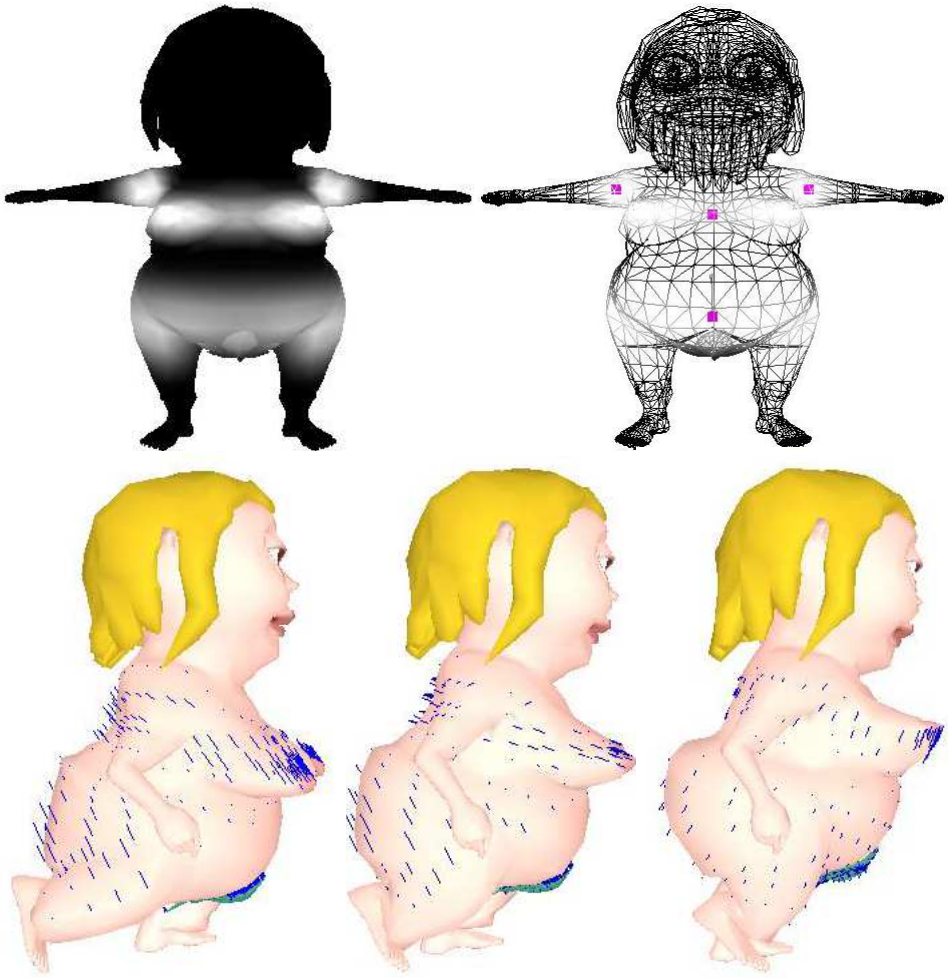
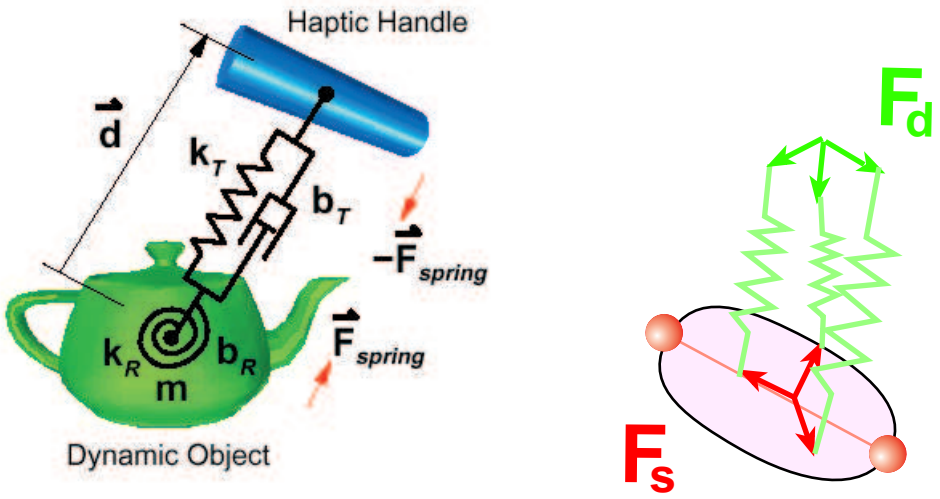


FIG. 4.21 – Le ventre, les fesses, les bras et la poitrine de Eve ont été animés en temps-réel par quatre éléments de chair connectés (utilisant donc quatres ressorts amortis). La première rangée d'images nous montre les poids de *skinning dynamique* associés aux ressorts ainsi que l'emplacement de ces derniers. La seconde rangée nous montre quelques images extraites de l'animation.

faudrait prendre en compte de la viscosité entre les ressorts), mais les temps d'exécution resteraient tout de même très faibles.

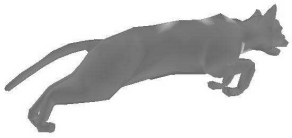


(a) (b)

FIG. 4.22 – Prise en compte de la torsion du ressort : (a)-première possibilité, on ajoute un couple de torsion au ressort amorti (image extraite de [MPT99]) ; (b)-seconde possibilité, on utilise trois ressorts amortis au lieu d'un seul.

Si l'on compare notre modèle animé par l'intermédiaire d'un ressort avec un modèle physique de type éléments finis résolu par analyse modale, la spécification d'un ressort avec ses paramètres associés correspond au mode principal de vibration de l'objet. En combinant plusieurs ressorts indépendants avec des paramètres de vibration différents, ceux-ci pourraient prendre en compte des modes de vibration supplémentaires. Ainsi, en ajoutant les vecteurs de position des ressorts avant d'effectuer le *skinning dynamique* il serait possible d'obtenir des vibrations plus réalistes et plus variées. De plus, doubler ou tripler le nombre de ressorts n'augmenterait les temps de calculs que faiblement et l'algorithme resterait ainsi temps-réel.

Concernant les fonctions d'atténuation utilisées, il est bien sûr possible d'explorer d'autres fonctions. Par exemple, on peut penser aux splines de Hermites qui décroissent avec des tangentes nulles et auxquelles il est possible de donner des formes variées. On pourrait laisser la possibilité à l'utilisateur de régler les autres paramètres de la courbe pour choisir différents modes d'atténuation. On pourrait aussi aller encore plus loin et laisser la possibilité à l'utilisateur de *dessiner* la fonction à condition de satisfaire la décroissance en 0 aux extrémités.



CHAPITRE 5

Plis dynamiques en temps-réel

Les animations de personnages que nous croisons continuellement au cinéma (Final Fantasy, Shrek) ou dans les jeux vidéos (Dawn, démos nvidia) manquent souvent de détails fins tel que les plis dynamiques de vêtements ou de la peau. Et ceci est d'autant plus vrai pour le jeu vidéo où les contraintes de temps-réel sont satisfaites au prix du réalisme.

Dans tout ce chapitre, nous employons le terme *dynamique* par opposition à *statique*, pour désigner les plis qui se créent pendant le mouvement et qui n'existent pas au repos[1]. Par exemple, si l'on regarde le modèle de la figure 5.1 (a), on peut voir un modèle $3D$ de bébé ne contenant qu'une grosse ride statique au niveau du ventre. Si ce personnage est animé par un *skinning* classique, peu importe le mouvement qu'il subit, aucune ride supplémentaire ne peut apparaître sur son t-shirt puisqu'elles ne sont pas définies dans le modèle initial. Cependant, si l'on s'attarde un instant sur la figure 5.1 (b), qui est une photographie réelle d'une chemise en mouvement, on voit immédiatement un grand nombre de plis, indispensables à la crédibilité de la scène. Ces plis sont *dynamiques* car ils sont créés, se déplacent et disparaissent tout au long du mouvement.

La raison pour laquelle de tels plis ne sont pas présents la plupart du temps dans les animations est que leur modélisation est pénible et très longue. En effet, si un infographiste souhaite les modéliser *à la main* sous Maya [ali], deux techniques s'offrent à lui :

- soit il définit des formes clefs de plis de différentes intensités correspondant à différentes poses du squelette, et ces formes sont ensuite mélangées pendant l'animation (voir section 2.3.1.1) ;
- soit il utilise des *deformers*, ce qui consiste à spécifier, pour chaque point ou groupe de points, une loi de déformation en fonction des angles des articulations par exemple.

Dans les deux cas, le processus de modélisation est long et compliqué.

[1] Attention, ici, le terme *dynamique* n'est pas employé dans le même sens qu'au chapitre précédent.

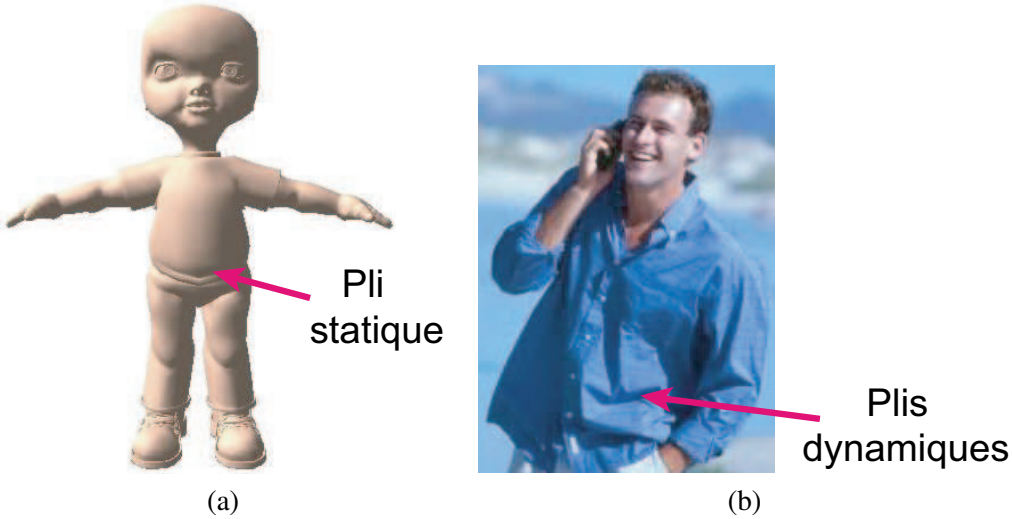


FIG. 5.1 – Des exemples de plis : (a)-un gros pli statique sur un personnage de synthèse ; (b)-des plis dynamiques réels.

C'est pourquoi la solution la plus largement utilisée pour produire les images de vêtements des films d'animation est d'avoir recours à des simulations physiques qui font apparaître des plis automatiquement. On pourra citer la simulation de la petite robe rose de Boo dans Monsters Inc. de Pixar qui a été entièrement calculée par modèle physique [BWK03]. Cependant, cette technique a deux inconvénients. Tout d'abord, en raison des temps de calcul très longs et des risques de divergence de la simulation, elle n'est pas utilisable dans les jeux vidéos. Deuxièmement, la simulation physique ne permet malheureusement pas à l'infographiste de préciser l'emplacement ou l'allure générale des plis, extrêmement difficiles à contrôler à travers les paramètres d'une simulation physique.

Ce chapitre, qui a fait l'objet d'une publication [LC04], a pour objectif de proposer un outil utilisable en temps-réel, permettant de spécifier localement l'emplacement d'une série de plis ainsi que la forme générale de ces derniers. Ainsi, au cours de l'animation, les plis apparaîtront et disparaîtront automatiquement dans la région choisie par l'artiste. Ces objectifs doivent de plus rester en accord avec ceux énoncés en introduction de la partie II, à savoir, les plis doivent pouvoir être ajoutés comme une couche supplémentaire au modèle et de façon simple et intuitive pour l'animateur.

5.1 Les plis dans la nature

Afin de créer un modèle adéquat, nous allons tout d'abord nous intéresser aux origines du phénomène de plissement. Que ce soit la peau ou bien la plupart des tissus, leur principale propriété est qu'ils sont peu compressibles. C'est pourquoi, pendant le mouvement, lorsque deux parties voisines de la peau ou d'un même vêtement se rapprochent, des plis se créent pour maintenir une surface constante. Ce phénomène est illustré sur la figure 5.2 qui montre des photos de plis réels de peau et de vêtements dans différentes situations.

FIG. 5.2 – Des plis dans la nature : de gauche à droite et de haut en bas, on peut voir un avant bras, le dos d'une main, un poignet et un t-shirt.

5.1.1 Propagation dans la zone de compression

Ces plis peuvent se créer de différentes façons. Par exemple, si la peau est pressée entre deux doigts comme sur la figure 5.3, les plis ont tendance à apparaître en même temps tout le long de la zone de compression et, plus la compression augmente, plus la hauteur des plis augmente. Si, par contre, la zone de compression est plus grande, on peut remarquer que les plis se forment d'abord près des extrémités, puis se propagent ensuite vers le milieu (voir figure 5.4). De la même façon, si la compression est créée par un seul doigt qui fait glisser la peau dans une certaine direction, les plis apparaîtront d'abord à l'extrémité où la force est appliquée, pour se propager ensuite. La figure 5.5 montre ce type de propagation sur de la peau (photo de gauche) et du tissu (photo de droite). Il est important de distinguer ces différents types de progressions tout en remarquant que dans tous les cas, les plis apparaissent dans une *zone de compression* délimitée par deux zones d'application de forces ou bien une zone d'application de force et une zone non déformée. Le lieu d'apparition des rides peut donc être encadré par deux zones extrémités qui marquent l'absence de plis (voir figures 5.3, 5.4 et 5.5).

D'autre part, la direction des plis qui se créent est entièrement déterminée par la direction de compression. Matérialisée par une double flèche sur les figures 5.3, 5.4 et 5.5, elle définit la direction des plis qui lui sont orthogonaux. Dans les cas simples, les plis formés sont de la même taille, espacés régulièrement et parallèles entre eux.

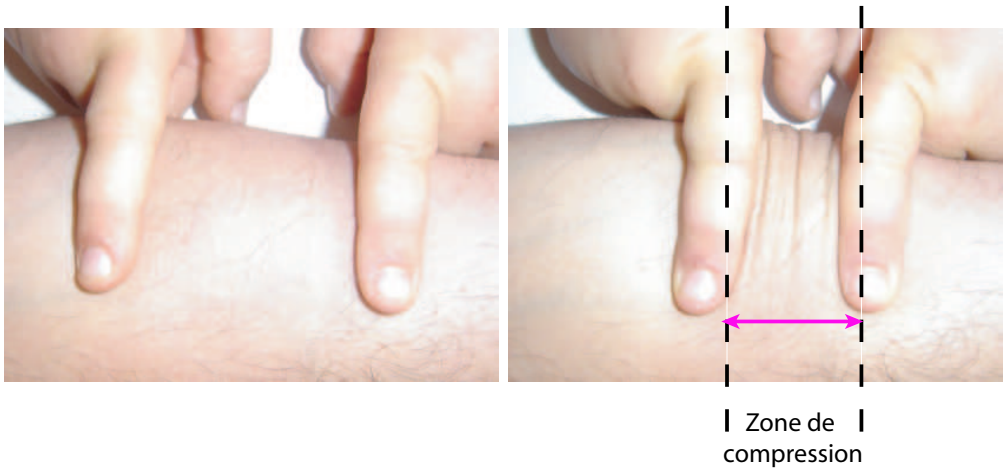


FIG. 5.3 – Apparition de plis simultanément tout le long de la zone de compression. Les plis se forment orthogonalement à la direction de compression.

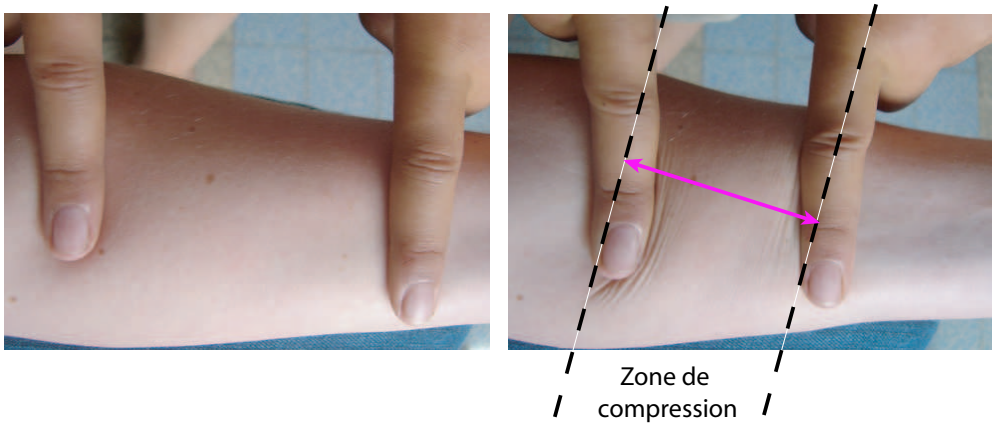


FIG. 5.4 – Apparition de plis auprès des zones d'application de forces puis propagation vers le milieu de la zone de compression. Les plis se forment orthogonalement à la direction de compression.

5.1.2 Atténuation

Par ailleurs la hauteur d'un pli pris individuellement n'est pas constante sur toute sa longueur. En effet, elle est maximale au centre alors qu'elle s'atténue jusqu'à devenir nulle à chaque extrémité. La figure 5.6 illustre ce phénomène. La largeur des plis, et donc leur atténuation plus ou moins régulière dépend de la largeur de la zone de compression.

5.1.3 Échelles

Si l'on observe des rides réelles comme celle de la figure 5.7, on remarque qu'elles sont souvent composées de rides à plusieurs échelles. Ce phénomène s'observe surtout sur la peau qui n'est pas un milieu homogène, la taille des plis de vêtements étant plus directement liée à l'épaisseur et à la rigidité du tissu. Nous souhaitons donc pouvoir obtenir ce genre de comportement multi-échelle avec notre outil.

FIG. 5.5 – Apparition de plis auprès de la zone d'application de la force puis propagation vers la zone non déformée. Les plis se forment orthogonalement à la direction de compression : (a)-sur de la peau ; (b)-sur du tissu.

FIG. 5.6 – Un gros pli réel : sa hauteur est maximale au centre et s'atténue vers les extrémités (image reproduite avec l'accord de Zabador).

5.2 Principe de la technique

Notre outil de création de plis permet d'ajouter des détails à une animation existante. Dans notre système, nous avons encore une fois utilisé un *skinning* classique pour calculer les déformations statiques du personnage. L'infographiste peut ensuite ajouter, avant ou pendant l'animation, des séries de rides dynamiques localement sur le personnage. L'animation de ces rides se fait automatiquement à partir de la position des points du maillage obtenus par le *skinning* à chaque pas de temps.

Le principe de notre technique est basé sur le fait que les rides ne se forment pas individuellement, et que dans chaque série de rides, l'orientation et la taille de chaque ride est similaire ainsi que l'espacement entre deux rides. D'autre part, si l'on fait des coupes transversales d'une série de rides, on peut remarquer que le profil est le même, quelle que soit la coupe. Seule l'amplitude des rides change : elle diminue au fur et à mesure que la coupe effectuée se rapproche des bords de la zone. Ce profil décrit la forme que prendrait un *fil* incompressible qui serait posé sur la zone de compression et attaché aux extrémités qui se rapprochent. Finalement, la seule donnée de la forme de ce *fil* peut être extrapolée pour obtenir la forme de tous les plis.

Nous avons donc décomposé notre algorithme dédié à la création de rides sur une surface $3D$ en deux étapes bien distinctes. La première consiste à créer une courbe se déformant dans un plan $2D$, servant de profil de rides en reflétant la déformation du *fil* et la seconde consiste à appliquer cette

FIG. 5.7 – Une main réelle montrant plusieurs rides à différents niveaux de détails.

courbe profil sur le maillage $3D$. Ce profil de ride, que nous appelons *courbe de contrôle* pour sa capacité à *contrôler* les déformations $3D$ du maillage, a pour propriété de pouvoir se déformer à longueur constante lorsque ses deux extrémités se rapprochent. Afin de garder une longueur constante, la courbe se voit contrainte à faire des plis. Ce sont ensuite ces plis de la courbe que l'on cherche à répercuter sur le maillage. Le modèle de courbe de contrôle peut donc être facilement changé pour permettre différentes sortes de plis. La section suivante décrit tout d'abord un modèle de courbe de contrôle très simple puis les paramètres modifiables qui permettent d'obtenir différents comportements de cette courbe en réaction à la compression. Elle est suivie d'une section nous montrant comment cette courbe peut contrôler la déformation du maillage sur lequel elle est appliquée.

5.3 Déformation de la courbe de contrôle

Les rides sont animées à l'aide d'une courbe de contrôle $2D$ discrète qui se déforme à longueur constante dans un plan orthogonal à la surface (voir figures 5.13 et 5.17). Elle est définie par ses deux extrémités, A et B, qui sont rigidement attachées à deux sommets du maillage. La longueur de la courbe, l, est égale à la distance entre les deux extrémités à l'instant où elle est définie. À cet instant, le maillage n'est pas forcément au repos, la longueur l de la courbe devant correspondre à son extension maximale, c'est à dire, au moment où les plis apparaissent et disparaissent. Ainsi, lorsque le maillage se déforme par *skinning*, les deux extrémités A et B se rapprochent ou s'éloignent et induisent une déformation de la courbe qui est contrainte à garder sa longueur constante. On peut assimiler cette courbe à un élastique linéaire de longueur au repos l qui s'étire lorsque A et B s'éloignent et qui plisse lorsque A et B se rapprochent.

5.3.1 Courbe de contrôle

Au repos, la courbe est un simple segment de droite de longueur l qui ne comporte aucun pli. Afin de lui permettre de se déformer, nous l'échantillonnons en n points de contrôle, répartis uniformément (voir courbe du haut sur la figure 5.8). Ces points de contrôle servent à définir la largeur et l'espacement des plis. La distance entre deux points consécutifs est appelé le pas de la courbe.

Nous allons illustrer le principe de l'apparition des bosses à l'aide du schéma de propagation présenté sur la figure 5.8, qui consiste à traiter les extrémités A et B différemment. L'extrémité B est fixe tandis que A se rapproche de B. Une première ride apparaît d'abord auprès de A, puis grandit, et enfin se propage quand elle atteint une certaine hauteur seuil. Les positions des points de contrôle de la courbe doivent donc être recalculés de façon à créer ces bosses au sein du profil de rides.

Il existe différentes stratégies de création de bosses. Néanmoins, l'algorithme de déformation reste le même.

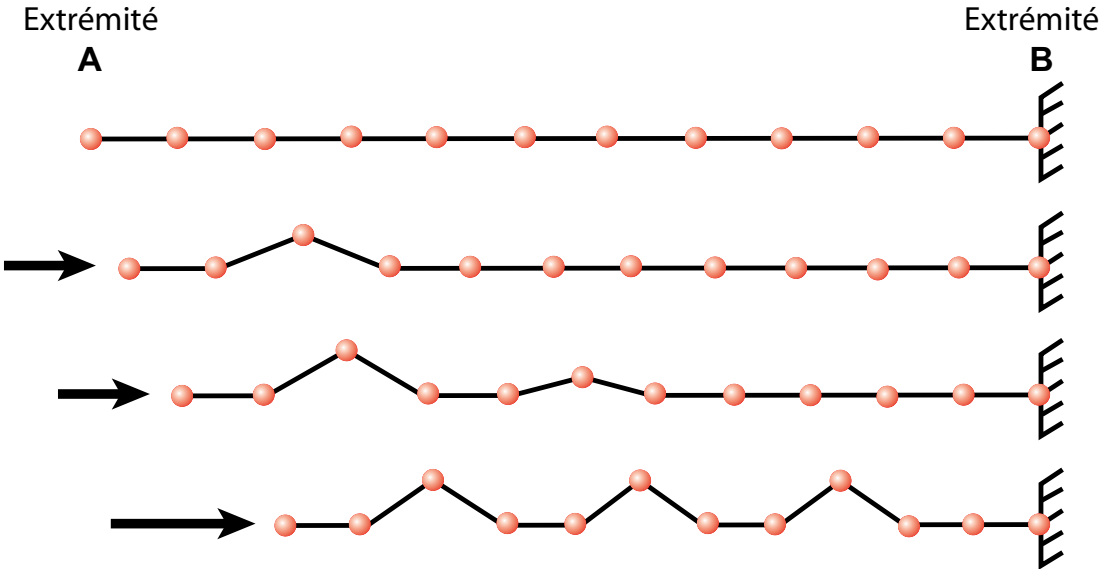


FIG. 5.8 – Une courbe de contrôle et ses déformations quand l'extrémité A se rapproche de l'extrémité B, fixe. Les bosses apparaissent d'abord auprès de A puis se propagent.

5.3.2 Algorithme de déformation

La déformation de la courbe est recalculée à chaque pas de l'animation. En effet, l'animation par *skinning* (ou toute autre méthode déformant un maillage) induit un changement de position des extrémités A et B nécessitant le repositionnement de tous les points de contrôle. L'algorithme se déroule de la manière suivante :

1. La nouvelle longueur du segment $\vec{AB}$ est calculée. On en déduit une nouvelle valeur du *pas* : $pas = \frac{||\vec{AB}||}{n-1}$.
2. Les points de contrôle sont réespacés sur le segment $[AB]$ suivant la valeur de pas. Ils restent donc également répartis sur le segment.
3. Certains des points de contrôle sont déplacés dans la direction orthogonale à $\vec{AB}$ et vers le haut afin de préserver la longueur totale de la courbe.

Le fait de répartir les points de contrôle uniformément sur le segment $[AB]$ permet d'assurer que chacun des plis générés du même niveau de détail aura la même largeur. C'est une considération importante qui traduit l'hypothèse que la peau ou le vêtement simulé est de composition homogène et d'épaisseur uniforme. Les points de contrôle qui se déplacent sont choisis en fonction du schéma de plis sélectionné par l'utilisateur. Cependant, afin qu'il soit possible de superposer plusieurs séries de rides, il est impératif que les extrémités A et B ne soient jamais modifiées par leur propre schéma de ride. Elles ne sont donc pas déplacées.

5.3.3 Conservation de la longueur

Afin de préciser et d'exprimer le principe de la conservation de longueur, nous introduisons un système de coordonnées $2D$ (nous verrons comment passer en $3D$ à la section suivante). Soit O, l'origine du repère, qui se trouve à l'extrémité A. $\vec{AB}$ nous donne la direction de l'axe des abscisses. L'axe des ordonnées est pris dans la direction orthogonale (voir figure 5.9).

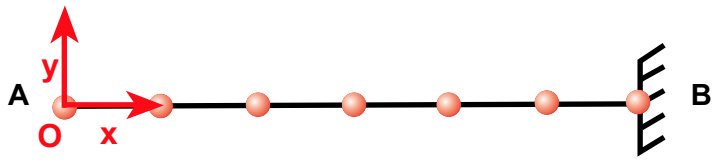


FIG. 5.9 – Système de coordonnées $2D$ associé à la courbe de contrôle. L'axe des abscisses est porté par le segment $\vec{AB}$ et l'axe des ordonnées lui est orthogonal.

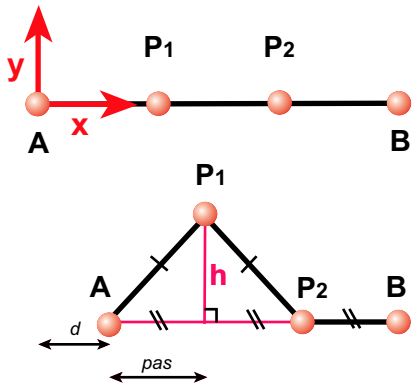


FIG. 5.10 – Le rétrécissement horizontal d est réinjecté en déplaçant P_1 dans la direction verticale d'une hauteur h.

Pour conserver la longueur totale, l'idée est de réinjecter, en déplaçant certains points d'une certaine hauteur sur l'axe des y, la longueur perdue horizontalement sur x lorsque A et B se rapprochent. Soit d le rétrécissement du segment $[AB]$ par rapport à sa position de repos ($d = ||\vec{AB}|| - l$) ; soit pas, la distance entre deux points de contrôle sur l'axe des abscisses ; et soit h, la hauteur que doit prendre le premier point de contrôle à être déplacé verticalement pour conserver la longueur totale de la courbe. Si l'on se réfère au schéma 5.10 qui représente la première bosse qui se forme sur la courbe de contrôle, A, B et P_2 sont des points de contrôle fixes, alors que $P1$ sera déplacé verticalement pour absorber la perte de longueur d. On veut alors s'assurer que $AP_1 + P_1P_2 = AP_2 + d$. Étant donné que les points sont également espacés, on peut facilement calculer h, la hauteur de P_1, en utilisant le théorème de Pythagore. On obtient :

$$h = \sqrt{d * pas + \frac{d^2}{4}}$$

Afin d'exprimer, de manière exacte, h en fonction de d et de pas, nous tirons avantage du fait que les points de contrôle sont espacés d'une distance égale sur l'axe des x.

5.3.4 Différentes stratégies possibles

Dans la pratique, la perte de longueur d est absorbée par plusieurs points de contrôle, ce qui peut se faire de plusieurs manières différentes. Comme nous avons pu l'observer à la section 5.1.1, les plis ne se créent pas et ne se propagent pas de la même façon dans toutes les situations. Nous proposons donc plusieurs algorithmes de propagation des bosses et quelques paramètres qui nous permettent, à partir de la même courbe de contrôle, de créer une grande variété de plis.

Nous avons isolé trois schémas principaux de propagation d'après nos observations :
- les bosses peuvent apparaître itérativement à partir de l'extrémité A du segment,
- symétriquement à partir des deux extrémités,
- ou bien de manière simultanée tout le long de la courbe.

Dans les deux premiers cas, l'utilisateur spécifie une hauteur de bosse maximale h_{max}. Lorsque cette hauteur est atteinte pour une bosse, les bosses se propagent. Dans le troisième cas, la longueur perdue d est divisée par le nombre de bosses et répartie équitablement entre toutes les bosses qui grandissent alors en même temps.

La forme de la courbe peut aussi être modifiée de façon très simple en changeant quelques paramètres comme l'espacement entre les bosses (zéro ou plusieurs pas au lieu d'un seul) ou bien la largeur des bosses. En effet, une crête peut être composée d'un ou plusieurs points de contrôle. Différentes formes de courbes de contrôle sont montrées sur la figure 5.11.

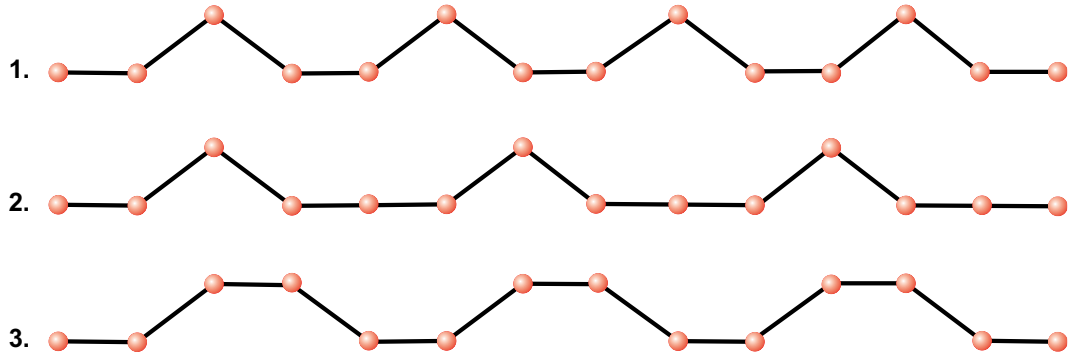


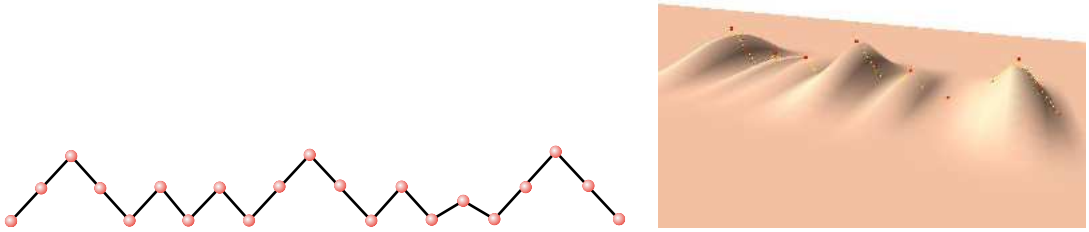
FIG. 5.11 – Différents schémas de courbes de contrôle : 1.-l'espacement entre les bosses est de 1 segment, la crête de chaque bosse est composée d'un seul point de contrôle ; 2.-l'espacement entre les bosses est de 2 segments ; 3.-la crête de chaque bosse est composée de 2 points de contrôle.

Tous les paramètres sont réglables par l'utilisateur pour laisser libre cours à sa créativité. Il peut ainsi choisir le modèle de propagation des rides, l'espacement, la hauteur et la largeur des bosses. Il est néanmoins imaginable de créer un schéma de ride aléatoire où les bosses pourraient apparaître au hasard le long de la courbe de contrôle, en respectant par exemple des zones de densité, un peu de la même manière que sont générées les cartes de territoire dans les jeux de stratégie. On pourrait aussi imaginer extraire ces paramètres à partir de photos ou de vidéos montrant des plis réels.

5.3.5 Niveaux de détails

Comme nous l'avons mentionné à la section 5.1.3 (voir figure 5.7), nous souhaitons pouvoir avoir des rides comportant plusieurs résolutions. Cette propriété doit être prise en compte au niveau de la courbe de contrôle. Pour cela, seulement un certain pourcentage de la longueur d est réinjecté dans la courbe au niveau grossier, le reste étant gardé pour les niveaux plus fins. Naturellement, pour obtenir des rides plus fines, la courbe de contrôle doit être subdivisée. Nous doublons simplement le nombre de points de contrôle à chaque raffinement, les nouveaux points se trouvant chacun au milieu de deux points de contrôle consécutifs. La figure 5.12 nous montre, en (a), la courbe de contrôle subdivisée, et, en (b), le résultat de son application sur un maillage simple. L'utilisateur peut donc spécifier le nombre de niveaux de détails souhaité, de nouveaux seuils pour la hauteur des rides, ainsi que le pourcentage de la longueur qu'il souhaite retrouver à tel ou tel niveau.

Il est important de remarquer que la spécification de la taille et du niveau de détail des rides



(a) (b)

FIG. 5.12 – Rides à différents niveaux de détails : (a)-une courbe de contrôle subdivisée ; (b)-application de cette courbe sur un maillage.

est réalisée indépendamment de la résolution du maillage sur lequel elles vont être appliquées. Au moment de la déformation du maillage, une subdivision locale détaillée à la section 5.4.5 est effectuée si nécessaire.

5.4 Déformation du maillage

Cette section décrit l'utilisation de la courbe de contrôle décrite précédemment pour déformer un maillage polygonal afin de créer la géométrie $3D$ des plis.

5.4.1 Mise en place de l'outil

La courbe de contrôle, comme son nom l'indique, contrôle directement la déformation du maillage. Pour qu'il se mette à faire des plis lorsqu'il se retrouve compressé dans une direction, il faut que la courbe de contrôle *suive* les déformations du maillage induites par le *skinning*. Pour cela, elle est ancrée à ce dernier au niveau de ses extrémités A et B.

L'infographiste, pour spécifier une série de rides, doit dessiner sur le maillage, à l'aide de la souris, un segment de droite qui correspond à la courbe de contrôle au repos. Les extrémités du segment sont attachées aux deux sommets les plus proches du maillage au sens de la distance euclidienne, l'extrémité A étant le premier point spécifié par l'infographiste, B le second. En effet, si l'utilisateur choisit un profil où les rides se propagent dans une seule direction, les extrémités A et B ne jouent pas le même rôle. La direction du segment détermine la direction des rides qui lui seront perpendiculaires, tandis que sa longueur nous donne la longueur au repos de la courbe, l.

Ensuite, l'infographiste peut régler les paramètres de la courbe comme le choix de la stratégie de propagation, l'espacement des rides, leur hauteur, et les niveaux de détails. Tous ces paramètres peuvent ensuite être modifiés interactivement, aussi bien pendant la phase de modélisation que pendant la phase d'animation.

Lors de l'animation, la courbe de contrôle se déforme dans le plan défini par la direction du vecteur $\vec{AB}$, désigné dans la suite par l'abscisse x, et la direction moyenne des deux normales à la surface aux extrémités A et B, désignée par l'ordonnée y dans la suite (voir figure 5.13). Ce système de coordonnées $3D$ correspond au système $2D$ de la figure 5.9 (même axe des x et même axe des y).

Le plan (x, y) dans lequel la courbe se déforme, la position des extrémités A et B de la courbe et donc sa longueur courante sont réévalués avant chaque rendu pour permettre à la courbe de contrôle de se déformer.

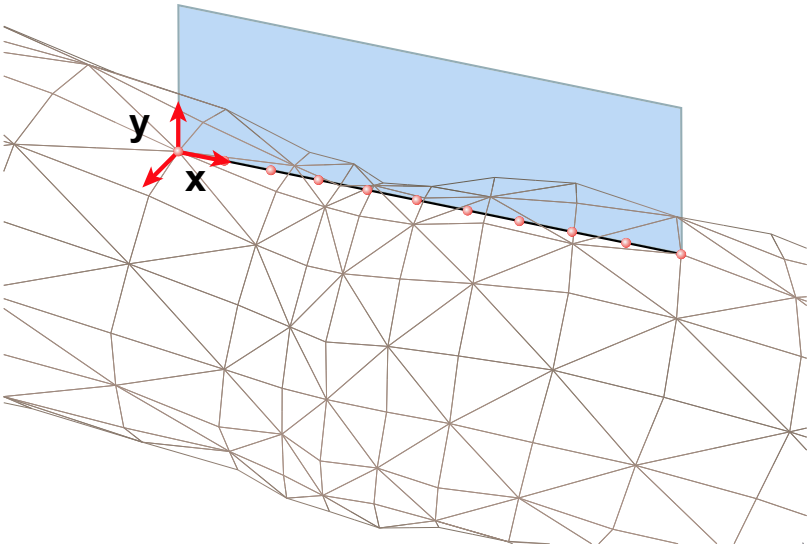


FIG. 5.13 – Plan de déformation de la courbe.

5.4.2 Région d'influence

Comme nous l'avons vu à la section 5.1, les plis apparaissent uniquement dans la zone de compression. Il va donc falloir préciser l'étendue de cette zone. Les sommets du maillage qui s'y trouvent seront alors influencés par la courbe de contrôle. Associée à chaque courbe, nous définissons une *région d'influence* qui s'étend de part et d'autre de la courbe. Concrètement, cela se présente sous la forme d'un rectangle centré sur le segment $[AB]$ dont la largeur est réglable par l'animateur. On peut voir une telle région d'influence sur la figure 5.14.

Les sommets du maillage qui seront influencés par l'outil et donc déplacés sont ceux qui se projettent dans le rectangle d'influence de l'outil, la projection se faisant selon la normale au plan du rectangle. Étant donné que le plan (x, z) peut croiser la surface (voir figure 5.14 (b)), cette recherche d'intersection doit se faire dans les deux sens de la normale. Cependant, cette sélection trop grossière ne permet pas d'éliminer les sommets qui se projettent dans le rectangle mais qui se trouvent de l'autre côté du maillage, sur la face non visible. Afin de les éliminer, nous ne gardons que les points dont la normale est dans le même sens et la même direction que celle du rectangle, à $\frac{\pi}{2}$ près. Il faut noter que cette condition n'est pas toujours suffisante, en particulier lorsqu'il y a des maillages constitués de plusieurs imbrications de sous maillages.

5.4.3 Atténuation aux bords de la région d'influence

L'utilisateur peut donc, en jouant sur la largeur du rectangle, définir la longueur des plis. En plus de ce paramètre, il peut choisir un profil d'atténuation des plis qui décrit comment leur hauteur décroît au fur et à mesure que l'on s'éloigne (sur z) de la courbe de contrôle, c'est à dire le long d'une ride. Le coefficient d'atténuation est toujours compris entre 1, au centre du rectangle, et 0, aux bords. Dans notre implémentation, l'utilisateur peut choisir de ne pas mettre d'atténuation, ce qui n'est pas très réaliste, ou bien, s'il en met, entre deux schémas d'atténuation différents.

Le premier schéma d'atténuation, représenté sur la figure 5.15, est linéaire ($y = 1 - abs(x)$) et très rapide à calculer mais n'a pas de tangentes nulles en -1, 0 et 1. Cela se traduit par des discontinuités aux bords du rectangle, c'est-à-dire entre la zone des plis et les zones sans pli.

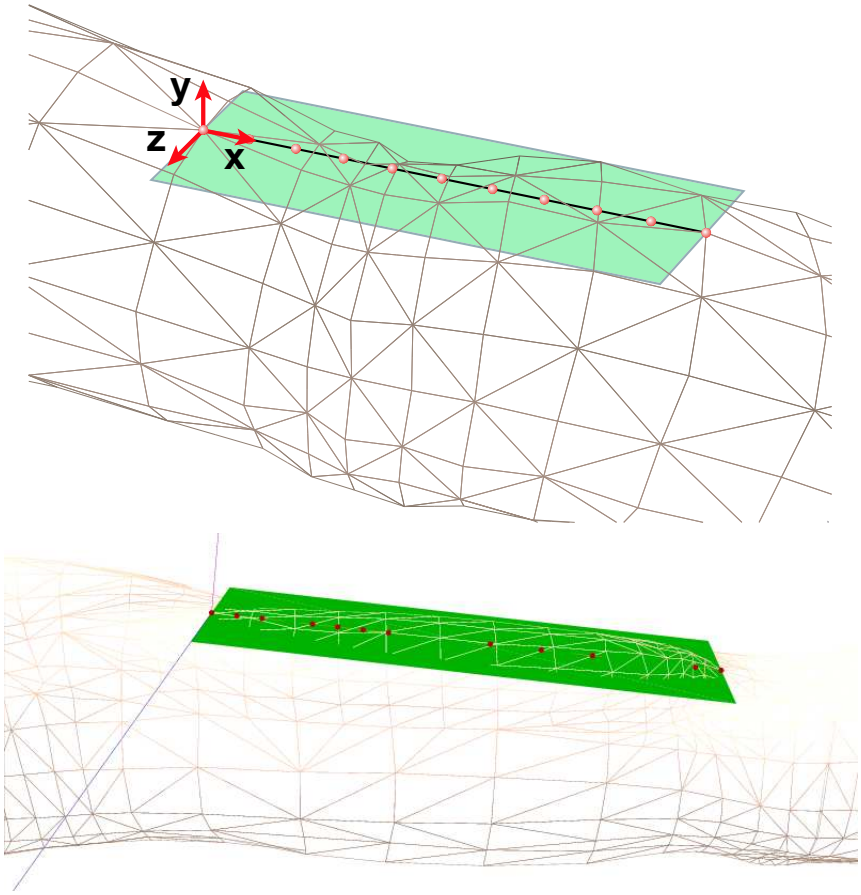


FIG. 5.14 – Une région d'influence est associée à chaque courbe de contrôle : (a)-la région d'influence et son système de coordonnées associé ; (b)-un cas simulé.

Le second utilise une fonction que nous avons déjà rencontrée au chapitre précédent : la fonction polynomiale de Wyvill, en forme de cloche ($y = 1 + \frac{(-4x^6+17x^4-22x^2)}{9}$). Cette fonction nous apporte les avantages d'une gaussienne (forme lisse, tangentes nulles en -1, 0, et 1) et est plus rapide à résoudre pour un x donné. En effet, la rapidité de résolution de l'équation est une donnée très importante à prendre en compte puisque le coefficient d'atténuation de chaque point de la zone d'influence doit être recalculé à chaque pas de temps.

Les effets de l'utilisation d'une fonction d'atténuation sont montrés sur la figure 5.16 où, pour les mêmes paramètres de la courbe de contrôle, on peut voir les plis sans atténuation (a), avec une atténuation linéaire (b), ou avec une atténuation non linéaire C^1 (c). Cette figure nous montre que seule la fonction de Wyvill produit des résultats naturels. Cependant, il est possible d'utiliser une autre fonction à condition qu'elle soit au moins C^1 et rapide à évaluer.

5.4.4 Déplacement des sommets influencés

Pendant l'animation du personnage, le squelette entraîne une déformation statique de la peau qui lui est attachée (*skinning*), et éventuellement des déformations dynamiques simulant la vibration de la chair (*skinning dynamique* introduit au chapitre 4). Le maillage est donc déformé en réponse au mouvement du personnage, et ces déformations sont des étirements du maillage à certains endroits

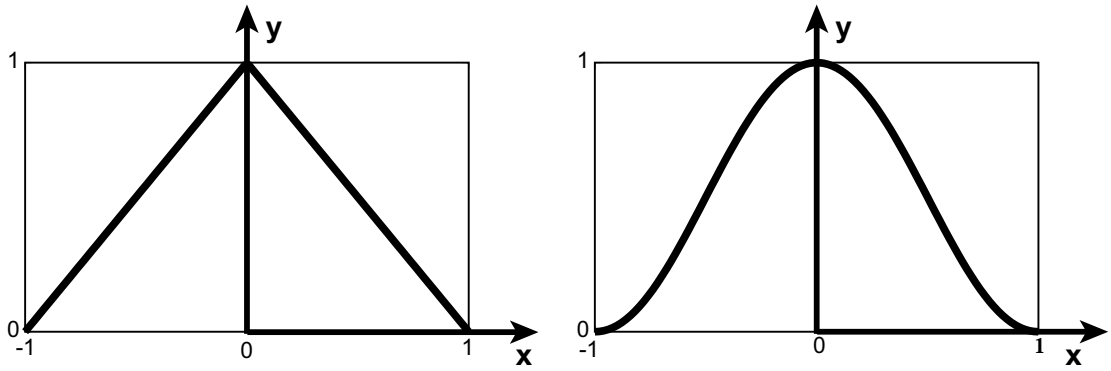


FIG. 5.15 – Un profil d'atténuation définit comment les bosses décroissent jusqu'à s'annuler aux bords du rectangle d'influence ; (a) $y = 1 - abs(x)$; (b) $y = 1 + \frac{(-4x^6+17x^4-22x^2)}{9}$.

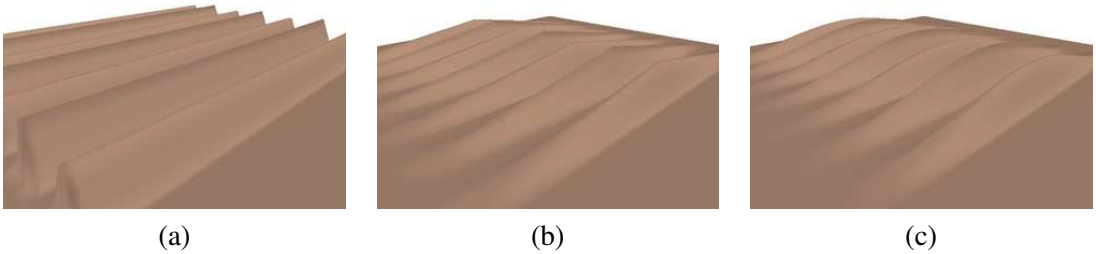


FIG. 5.16 – Mise en évidence de la nécessité d'utiliser une fonction d'atténuation continue C^1 : (a)- pas d'atténuation ; (b)-atténuation linéaire ; (c)- atténuation non linéaire utilisant la fonction de Wyvill.

et des compressions du maillage à d'autres. Or, si la peau ou certains vêtements s'étirent bien par rapport à leur position au repos, il sont quasiment incompressibles. Ceci se traduit par la formation de plis. Il s'agit donc pour nous, connaissant le maillage déformé après les diverses étapes de *skinning*, de détecter les endroits où le maillage s'est retrouvé compressé et d'ajouter à ces endroits, les rides au préalable spécifiées par l'infographiste.

À chaque pas d'animation, il faut, pour chaque sommet du maillage influencé par l'outil, calculer le déplacement qu'il doit subir. Il est obtenu par les coordonnées (x, z) de la projection du point sur le rectangle d'influence selon la normale de celui-ci. La coordonnée x donne la hauteur h du point sur la courbe de contrôle, et la coordonnée y décrit l'atténuation a subie par ce point au fur et à mesure qu'il s'éloigne du centre du rectangle. Le schéma 5.17 nous montre l'obtention de h et a pour un point P quelconque.

Concrètement, l'élévation finale du point est sa hauteur h, multipliée par l'atténuation a :

$$elevation = h * a$$

Cette élévation est appliquée au point, dans la direction de sa normale à la surface avant application de la ride. Ce choix est motivé par plusieurs raisons : tout d'abord, le fait de prendre la normale au point avant déformation réduit fortement les risques d'auto-collision. Ensuite, étant donné que plusieurs schémas de rides peuvent se superposer, si l'on utilisait la normale à la surface courante, la hauteur

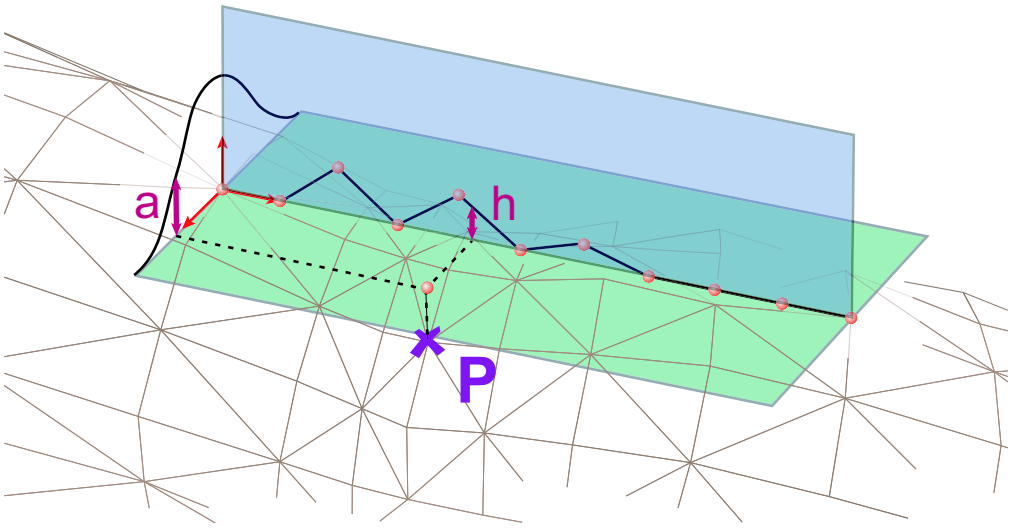


FIG. 5.17 – Calcul de l'élévation d'un sommet influencé : l'abscisse de la projection du point P dans le rectangle d'influence donne la hauteur h qui est atténuée par a calculée à partir de son ordonnée.

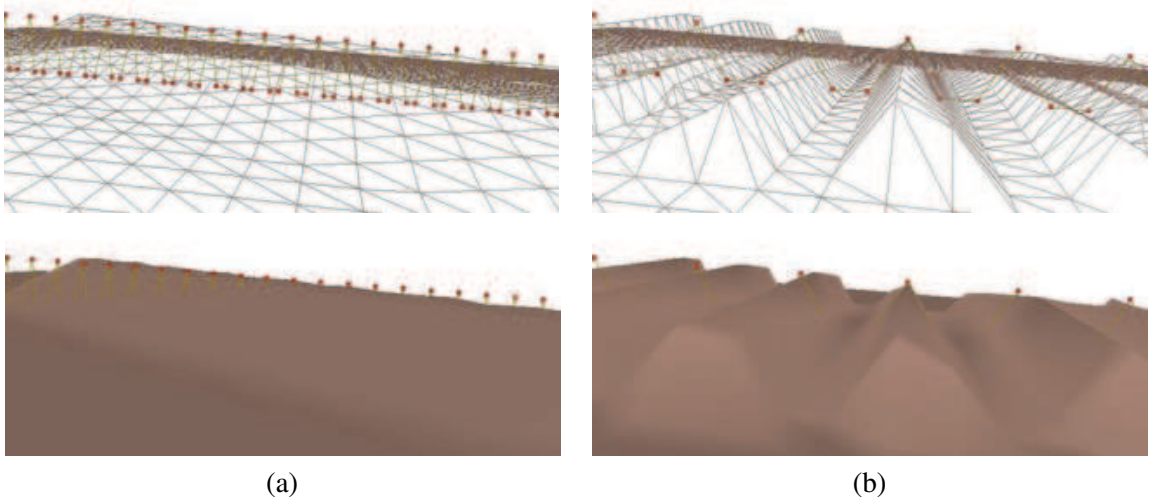
ajoutée en réponse à la seconde ride serait ajoutée dans une direction qui n'aurait aucun sens pour un point se retrouvant dans une montée ou une descente de bosse.

5.4.5 Subdivision du maillage

Pour obtenir de bons résultats, cette méthode nécessite un maillage suffisamment échantillonné par rapport à la courbe de contrôle pour que les rides puissent s'exprimer. En effet, il faut que les côtés des triangles du maillage soient plus petits que l'espacement sur x de deux points de la courbe de contrôle. Ceci est illustré sur la figure 5.18 où l'on montre la différence entre un cas qui fonctionne (b) et un cas qui ne fonctionne pas (a).

Demander à l'infographiste de raffiner *à la main* le maillage dans les zones où il place les rides serait complètement à l'encontre du principe de la technique qui se veut simple et non pénible à utiliser. En effet, cela forcerait l'animateur à prévoir l'emplacement des plis potentiels au préalable, ce qui serait largement aussi pénible que de les modéliser à la main, à l'aide de formes clefs. L'autre inconvénient majeur serait que l'animation en elle-même se trouverait ralentie à cause d'un nombre de polygones élevé, aussi bien au niveau du *skinning*, de l'ajout des effets dynamiques ou du traitement des collisions. Enfin, cela forcerait aussi à utiliser un maillage fin pour le rendu, et ceci même lorsque les rides ne seraient pas visibles (cas où la courbe de contrôle n'est pas compressée à ce moment de l'animation par exemple).

Nous avons donc choisi de raffiner le maillage localement, uniquement lorsque ceci est nécessaire. Ainsi, lorsque la zone des rides est en étirement, le maillage n'est pas subdivisé. D'autre part, lorsqu'un triangle est recouvert par plusieurs rides, nous subdivisons au niveau le plus fin nécessaire. L'algorithme de subdivision que nous utilisons est un *butterfly modifié* (voir état de l'art, section 2.1.3). En effet, cet algorithme a l'avantage d'être interpolant, ce qui nous évite de recalculer un certain nombre de propriétés des points déjà existants telles que la position, la normale, ..., mais est aussi applicable directement sur un maillage triangulaire. Cependant, comme nous ne subdivisons pas *tous* les triangles du maillage, nous devons avoir recours à un artifice pour les triangles qui se situent à la jonction d'une zone subdivisée et d'une zone non subdivisée. Actuellement, ces triangles sont sim-



(a) (b)

FIG. 5.18 – Importance du rapport entre le *pas* de la courbe de contrôle et la résolution du maillage : (a)-la courbe de contrôle est beaucoup trop échantillonnée par rapport au maillage, une seule grosse bosse apparaît au lieu d'une série de rides très fines ; (b)-le maillage est beaucoup plus fin que le *pas* de la courbe, les plis $3D$ reflètent correctement la déformation de la courbe de contrôle.

plement coupés en deux (ou 4, 6, ... suivant le nombre de subdivisions) ce qui résulte en la création de nombreux triangles très étirés (voir figure 5.19). La solution, non implémentée, serait de propager la subdivision aux triangles voisins en baissant le nombre de subdivisions de 1 à chaque fois, ce qui nous donnerait, au pire, des triangles coupés en deux.

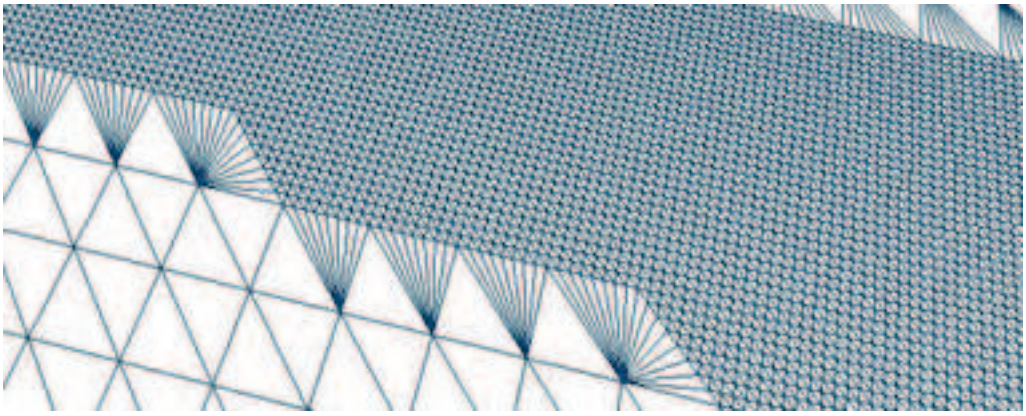


FIG. 5.19 – Un exemple de maillage subdivisé localement à un niveau 3. On peut remarquer les triangles très étirés au bord de la zone subdivisée.

Il faut aussi prendre en compte que c'est l'étape de subdivision, effectuée à chaque rendu, qui est la partie coûteuse de l'algorithme, tout simplement car la recherche du voisinage des triangles n'est pas trivial. Nous gardons du temps-réel, voire interactif, car le nombre d'étapes de subdivision est généralement petit (1 ou 2, sachant qu'à chaque subdivision, la taille des triangles est divisée par 4). Un bon espoir cependant est la naissance de la programmation d'algorithmes de subdivision sur les cartes graphiques [BS04], ce qui pourrait considérablement réduire les temps de calcul.

5.4.6 Plis superposés

Des formes de plis plus compliquées et variées peuvent être obtenues en combinant plusieurs séries de rides. Pour cela, on peut superposer différents outils. C'est par exemple le genre de rides que l'on observe sur les pantalons au niveau du genou. La figure 5.20 nous montre deux séries de rides superposées utilisées pour obtenir des plis plus naturels sur un pantalon virtuel.

FIG. 5.20 – Superposition de plusieurs séries de rides sur l'arrière du genou d'un pantalon virtuel. Lorsque le genou se plie, les rides apparaissent.

Dans ce cas, l'utilisateur dessine plusieurs profils de rides, qui se croisent. Lors du calcul des déplacements des sommets influencés, l'élévation imposée aux sommets est la somme des élévations provoquées par chaque profil de plis. Néanmoins, il est tout à fait possible de laisser libre choix à l'utilisateur d'additionner ou de moyenner ces élévations pour obtenir l'élévation finale.

5.5 Résultats et performances

Tous les résultats présentés dans cette section ont été réalisés en temps-réel sur un ordinateur standard équipé d'un processeur AMD Athlon XP 1700+, d'une RAM de 1 Giga et d'une GeForce4 Ti.

5.5.1 Rides du front

Notre premier exemple illustré sur la figure 5.21, montre l'utilisation de notre outil sur les fronts de deux modèles de tête différents : un homme sur la rangée du haut et une femme sur celle du bas.

Étant donné que nous n'avions pas de logiciel d'animation faciale à notre disposition, nous avons manuellement changé la longueur au repos de la courbe de contrôle pour simuler une compression du maillage. Les résultats obtenus, pour différents paramètres de la courbe de contrôle, sont montrés sur la figure 5.21.

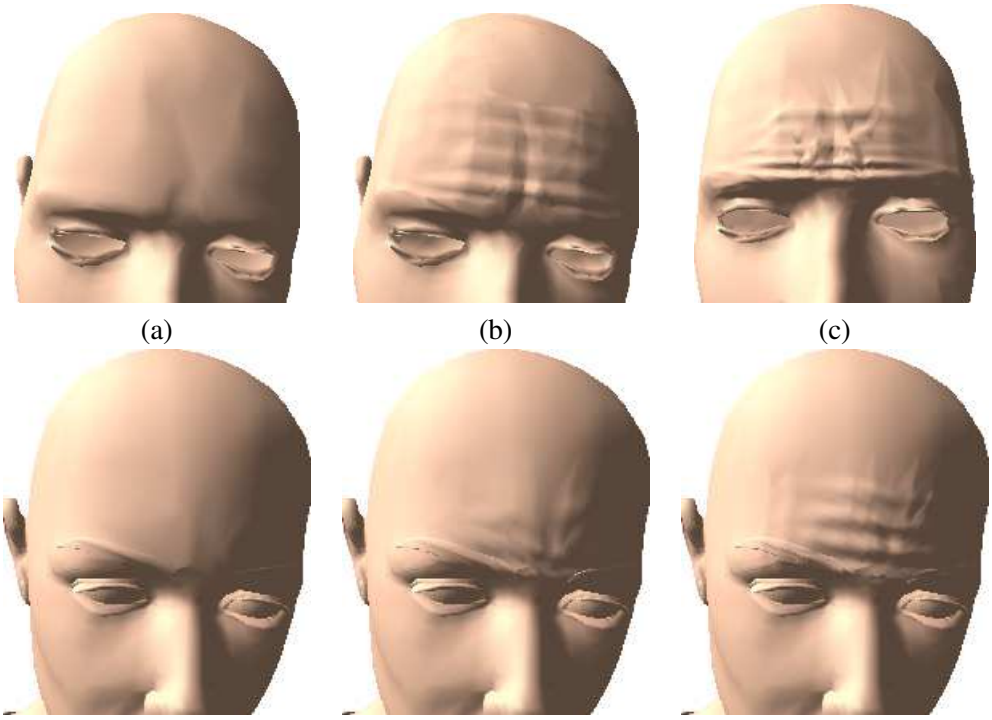


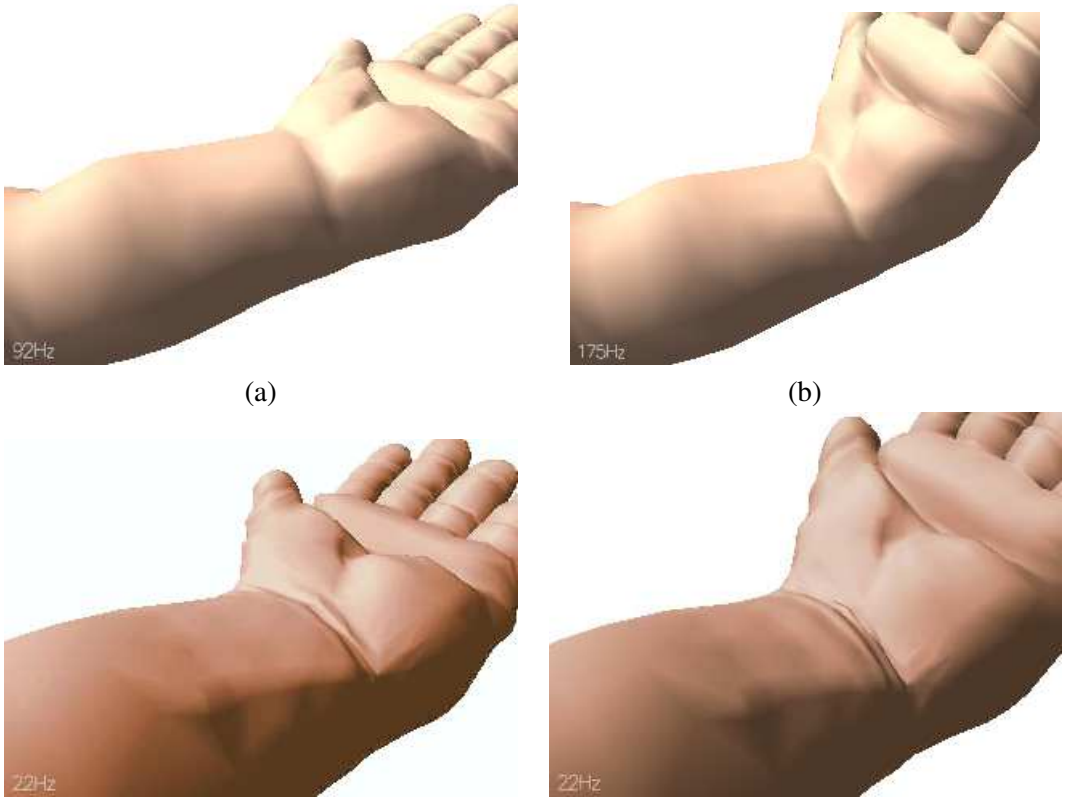
FIG. 5.21 – Utilisation de notre outil pour créer des rides sur les fronts de deux modèles différents : un modèle d'homme pour la rangée du haut et de femme pour la rangée du bas. (a) et (d)-maillages non déformés ; (b), (c), (e) et (f)-maillages déformés avec différents paramètres.

5.5.2 Plis au niveau des articulations

La principale cible de notre outil est la modélisation des plis qui se créent au niveau des articulations lorsque celles-ci plient. La figure 5.22 montre les résultats que l'on obtient sur l'animation d'un poignet. Le modèle est animé par *skinning* classique. La rangée du haut montre les déformations dues au *skinning* seul, alors que la rangée du bas nous montre l'effet de notre outil. Sur ce modèle composé de 2480 polygones, nous avons utilisé une seule courbe de contrôle. L'animation se déroule à 23 frames par seconde lorsque le maillage se subdivise à deux niveaux, et à 188 frames par seconde lorsque le maillage n'est pas subdivisé, c'est à dire quand la courbe de contrôle n'est pas active.

5.5.3 Plis de vêtements

Enfin, notre dernier exemple illustre l'utilisation de notre outil sur des vêtements (voir figure 5.23). Nous avons placé 4 courbes de contrôle sur notre modèle, deux sur l'avant du genou (rangée du bas), et deux qui se recouvrent sur l'arrière (rangée du haut). L'animation tourne à 17 frames par seconde lorsqu'un niveau de subdivision est actif pour un maillage composé de 6362 polygones. Le frame rate descend malheureusement à 7.5 frames par seconde lorsque l'on modélise des plis plus fins qui nécessitent deux niveaux de subdivision.



(a) (b) (c) (d)

FIG. 5.22 – Illustration de l'utilisation de notre outil sur un poignet de bébé. (a) et (b)-animation standard n'utilisant que le *skinning* classique : pas de rides présentes ; (c) et (d)-notre outil fait apparaître des plis dans le creux du poignet pendant le mouvement.

5.6 Limitations et extensions de l'algorithme

Avant de conclure, nous souhaitons proposer deux extensions à notre algorithme, qui sont encore en cours de développement et pour lesquelles nous n'avons malheureusement pas encore de résultats.

5.6.1 Courbe de contrôle multi-résolution

La structure actuelle de notre outil nous permet aisément de remplacer notre courbe de contrôle par une autre. Le principal inconvénient de la courbe utilisée actuellement est qu'elle *ne colle pas* au maillage. En particulier, quand elle est employée au creux d'une articulation, elle approche assez mal la conservation d'aire de la surface au dessus de laquelle elle est animée. D'autre part, les plis obtenus sont tous très semblables, et régulièrement espacés. Même si l'on introduisait une apparition aléatoire des bosses, celles-ci resteraient de même forme et de même amplitude. Or, si ce modèle convient bien aux matériaux homogènes, il est moins adapté à la simulation de zones de peau non homogènes comme le creux de la main ou les peaux âgées par exemple.

Pour remédier à ces deux problèmes, nous souhaitons utiliser une courbe qui suit le maillage et qui se déforme de façon très naturelle (pas de symétries). Pour cela, des travaux ont été entrepris en remplaçant notre modèle de courbe par celui de Sauvage et al. [SHB04]. Ce modèle de courbe

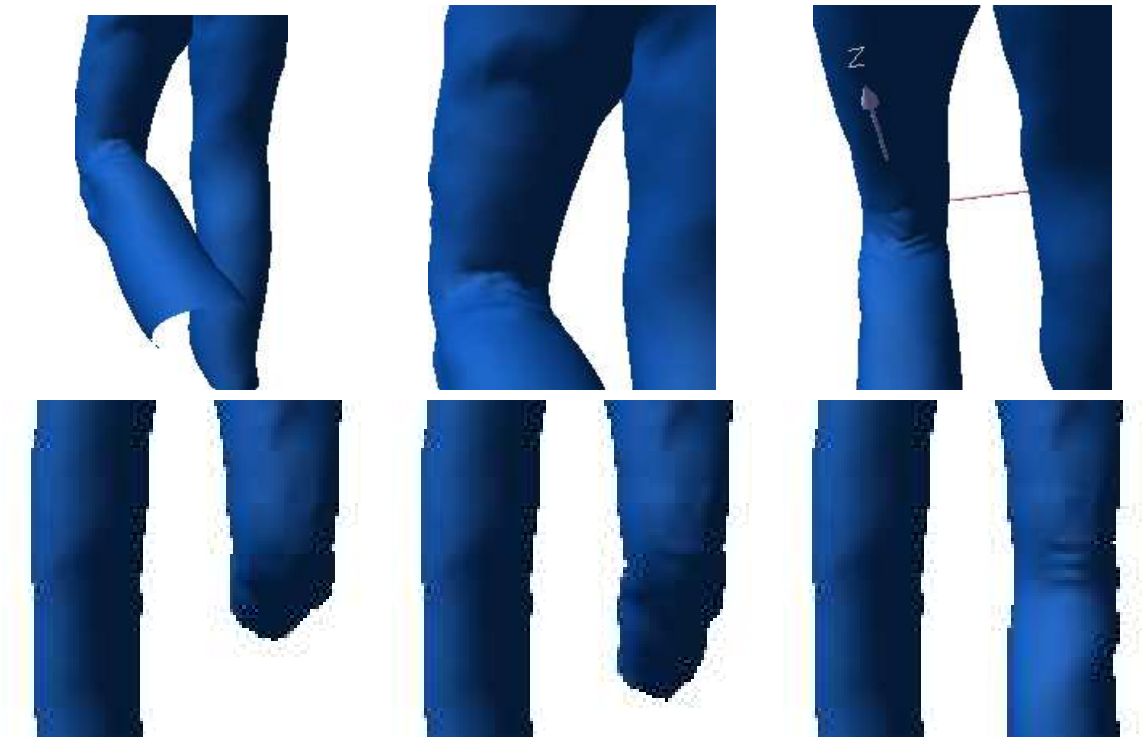


FIG. 5.23 – Illustration de notre outil de création de plis sur un pantalon virtuel. La ligne du haut montre l'arrière d'un pantalon où les plis sont créés par deux séries de plis superposées. La ligne du bas montre l'avant d'un pantalon où une série de rides est disposée sur la jambe de droite. On peut constater la différence avec la jambe de gauche, dépourvue de plis.

se déforme à longueur constante, en temps *presque* réel et est multi-résolution. Cela nous permet d'utiliser les mêmes points d'ancrage que nous avons déjà pour définir notre courbe actuelle, et de rajouter des points sur la courbe en projetant cette courbe sur la surface. De cette façon, au repos, la courbe est entièrement alignée sur la surface $3D$. Pour garder sa longueur totale constante, cette courbe tend à garder une longueur constante entre deux points consécutifs de la courbe et le fait en plissant aux endroits *prédisposés* du maillage. Nous avons bon espoir d'obtenir des résultats plus réalistes et à surface un peu plus exactement constante.

5.6.2 Rides courbes

L'autre axe de recherche que nous avons commencé à explorer est la possibilité d'obtenir des rides courbes. Ce sont typiquement les rides que l'on observe quand on pousse la peau avec un doigt. Pour cela, nous souhaitons ajouter un paramètre de courbure à la région d'influence, qui, de ce fait, ne serait plus rectangle.

L'autre aspect de cette application est l'interface qui est nécessaire pour spécifier un point d'application de force ainsi qu'une direction et une intensité. Admettons que nous pouvons déterminer un point d'impact, une direction de poussée et une intensité, il est alors possible d'attacher au maillage une de nos courbes de contrôle. En effet, dans ce cas, l'extrémité A n'est rien d'autre que le point d'impact, et, l'extrémité B peut être choisie sur le maillage, dans la direction de la force et suffisamment loin pour être sur une partie du maillage non déformée.

Un fois la courbe spécifiée de cette façon, il faut bien sûr *compresser* le maillage afin que notre outil puisse répondre à la compression en créant des plis. Pour cela, nous attachons une sphère gaussienne au point d'impact [PP01], de rayon inférieur à la distance entre les deux extrémités de la courbe (et d'une valeur qui doit aussi dépendre de l'intensité de la poussée). Cette sphère a pour but de spécifier un coefficient de déplacement de chaque sommet qui se trouve dans la sphère, variant de 0 à 1, pour la partie du maillage en arrière qui doit être étirée, et de 1 à 0, pour la partie qui doit être compressée, et qui donne directement la pondération du vecteur de déplacement du doigt qu'il faut associer à chaque point du maillage situé dans la sphère.

Enfin, il faudrait que cette déformation soit réversible pour que quand l'utilisateur lâche la souris, le maillage reprenne plus ou moins lentement sa position initiale.

5.7 Conclusion et travaux futurs

Nous avons présenté un outil géométrique permettant de créer des plis en temps-réel sur une surface polygonale quelconque. Cette surface doit être préalablement animée par *skinning* et notre outil agit dans les zones où le maillage se trouve compressé en créant des plis. Plusieurs paramètres sont à la disposition de l'utilisateur afin qu'il puisse régler la fréquence des plis, leur forme, leur hauteur ainsi que leur largeur. Ces paramètres sont intuitifs, et faciles à régler, le retour visuel étant disponible en temps-réel.

À plusieurs passages, au cours de ce chapitre, nous avons évoqué des possibilités d'extension de ce travail. En particulier, des améliorations peuvent être apportées au niveau de la courbe de contrôle et de la répartition des bosses, au niveau du choix de la fonction d'atténuation ou de la manière de prendre en compte les effets combinés de deux schémas de rides superposés, le principe de la technique restant le même.

Outre ces petites améliorations possibles, un travail plus conséquent doit être fourni en ce qui concerne la subdivision locale du maillage, qui est le goulot d'étranglement de l'algorithme, en ce qui concerne les performances. Pour cela, il est possible de réduire le parcours, pour le moment exhaustif, des triangles du maillage en utilisant un découpage du maillage, en octree par exemple.

Par ailleurs, l'algorithme actuel ouvre la voie à de nombreuses extensions de l'algorithme en ce qui concerne la région d'influence de l'outil combinée avec sa fonction d'atténuation. La première, mentionnée à la section précédente, pourrait être d'ajouter un paramètre de courbure à cette région d'influence afin de permettre des rides courbes. Une autre possibilité serait d'utiliser une région d'influence non planaire, mais cylindrique par exemple, pour pouvoir notamment simuler des plis qui s'enrouleraient autour du bras.

À plus long terme, nous aimerions ajouter un système de détection de collisions et d'auto-collisions ainsi qu'une réponse adéquate de la déformation.

Enfin, des possibilités de travail futur s'ouvrent aussi au niveau de l'implémentation de notre outil. Une des motivations principales pour la structure de notre algorithme et notamment la subdivision à la volée, est que nous souhaitons implémenter cet outil sur carte graphique. L'algorithme, ne modifiant finalement que la position des sommets du maillage, pourrait être aisément implémenté dans un *vertex program*. Le seul obstacle actuel est encore une fois, la subdivision, puisqu'elle nécessite que chaque sommet du maillage ait connaissance de son voisinage ce qui n'est pas directement possible lorsque les données se trouvent sur la carte graphique. Néanmoins, l'intérêt et les motivations sont bien présentes, notamment pour utiliser cet outil dans des jeux vidéos. Une seconde implémentation que nous aimerions proposer est un plug-ins pour Maya [ali]. Cela permettrait aux infographistes

d'utiliser notre outil et de nous donner, en retour, ses inconvénients ou améliorations possibles du point de vue de l'utilisateur expert en modélisation/animation.

CHAPITRE 6

Combinaison des deux outils

Ce chapitre se propose de montrer que les deux techniques présentées aux chapitres précédents sont totalement compatibles et utilisables conjointement (le lecteur pourra aussi se reporter à la publication correspondante [LCA04]). Pour cela, nous avons repris le modèle du chat animé par *skinning dynamique* présenté au chapitre 4.

6.1 Résultats

Nous avons ajouté deux séries de plis sur l'un des côtés du chat, l'une reliant le haut de la patte avant au ventre, et l'autre reliant le haut de la patte arrière au ventre (voir figure 6.1). Les plis étant difficilement visibles sur des images réduites fixes, la figure 6.2 montre la déformation qui apparaît au niveau de la patte avant du chat lorsque celle-ci se rapproche du ventre, ainsi que le maillage raffiné et déformé, le tout en vue de dessous.

Les résultats obtenus pour une animation complète sont montrés sur la figure 6.3. La figure 6.4 montre la même animation en fil de fer. On peut observer le mouvement du ressort ainsi que le raffinement à la volée du maillage lorsque les plis sont actifs.

6.2 Comparaison avec un *skinning* simple

Afin de montrer l'apport de l'ajout de détails par nos outils sur l'animation du chat, nous avons comparé les résultats obtenus pour un *skinning* simple avec ceux obtenus pour un *skinning dynamique* uniquement d'une part, avec des plis dynamiques uniquement d'autre part et enfin avec un modèle complet incorporant *skinning dynamique* et plis dynamiques. La comparaison est effectuée sur la figure 6.5 pour deux poses différentes de l'animal. La rangée du haut correspond à l'animation par

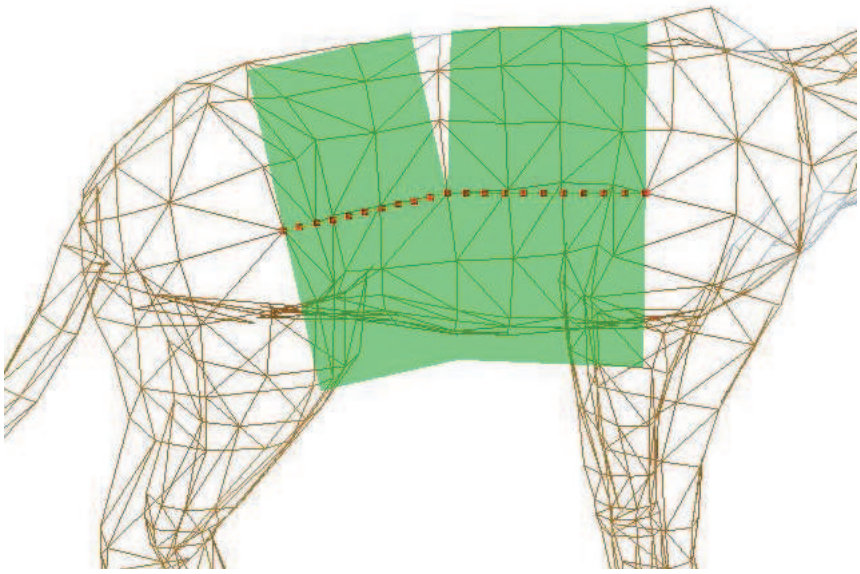


FIG. 6.1 – Ajout de deux séries de plis sur le modèle de chat : une série entre la patte avant et le ventre, et une seconde série entre la patte arrière et le ventre.

skinning classique founie par l'infographiste. La seconde rangée montre l'ajout de notre *skinning dynamique* sur la région du ventre. La troisième rangée montre l'ajout de plis sur le *skinning* classique. Et enfin, la dernière rangée nous montre la combinaison des deux outils.

6.3 Comparaison avec une situation réelle

Enfin, afin de valider notre modèle, nous avons comparé nos résultats à une image réelle de lynx. Comme on peut le voir sur la figure 6.6, des plis se forment au niveau de la patte avant lorsque l'animal se déplace (entourés d'une ellipse). La figure 6.7 montre le même type de déformations obtenues avec nos outils utilisés conjointement en temps-réel sur une animation existante.

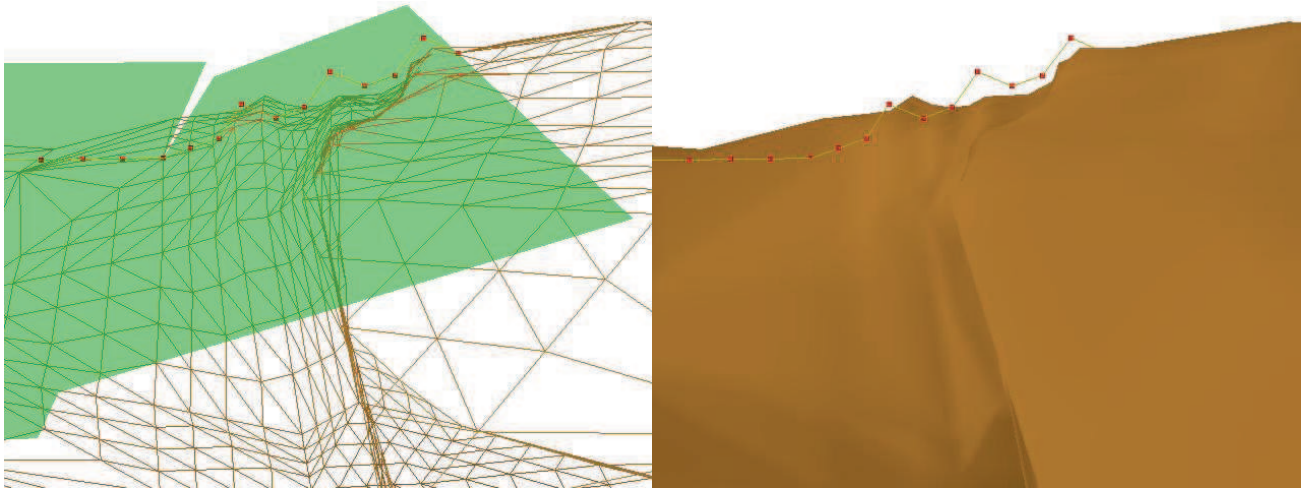


FIG. 6.2 – Lorsque le chat se déplace, des plis se forment au niveau de la patte avant. Cette image montre une vue rapprochée de dessous de cette zone.

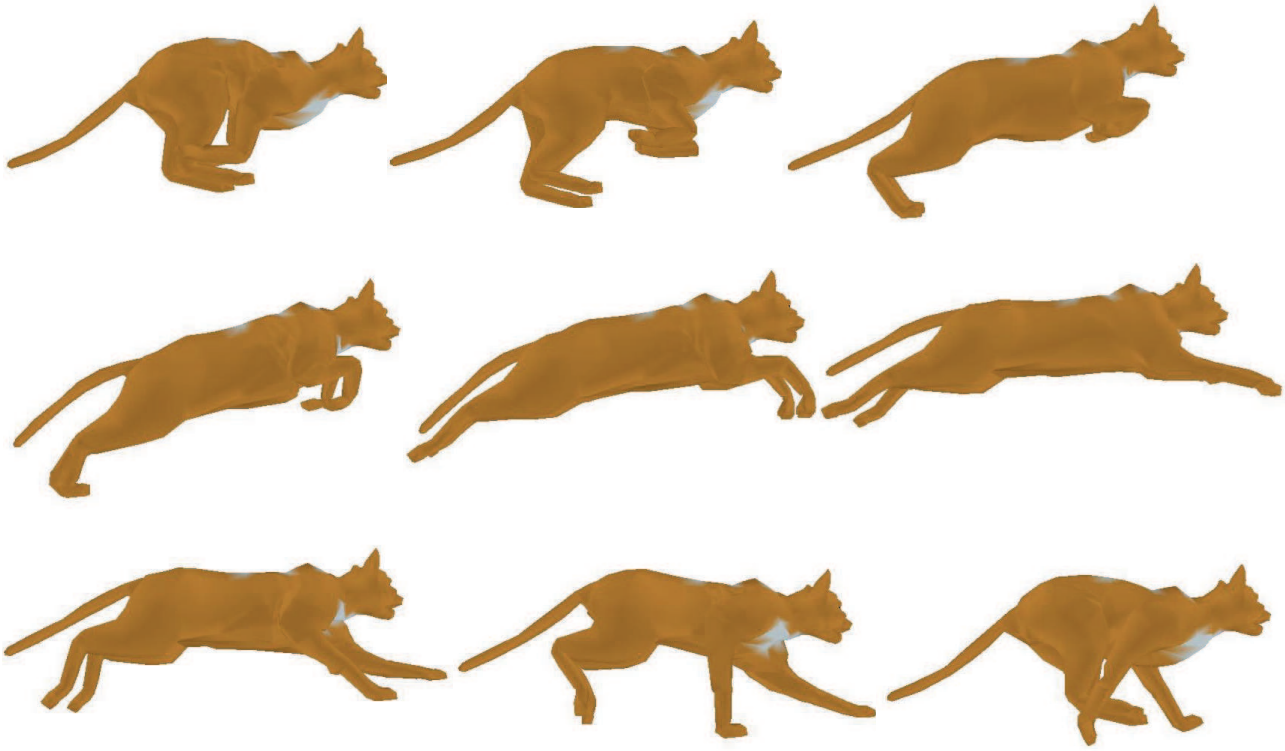


FIG. 6.3 – Animation du modèle du chat sur laquelle des effets dynamiques et des plis dynamiques ont été ajoutés.

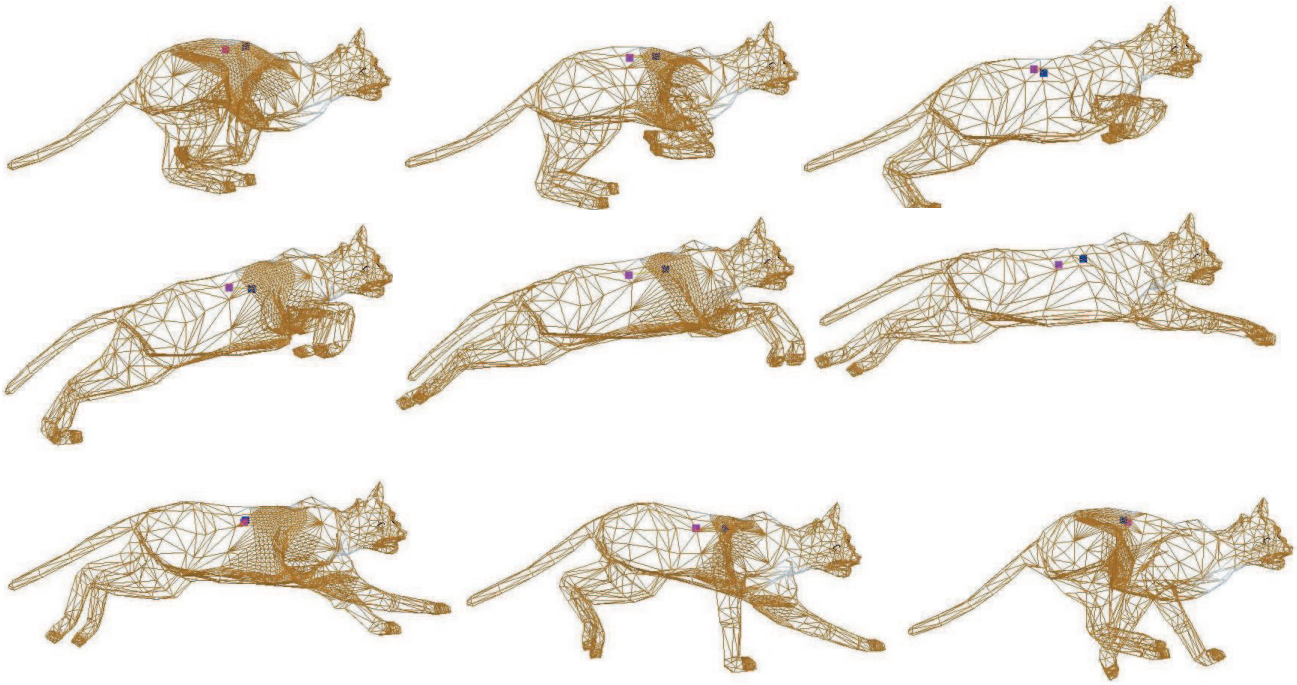


FIG. 6.4 – Même animation que sur la figure 6.3, affichage en fil de fer. Noter le mouvement du ressort dynamique ainsi que le raffinement du maillage à la volée, uniquement lorsque les plis dynamiques sont actifs.

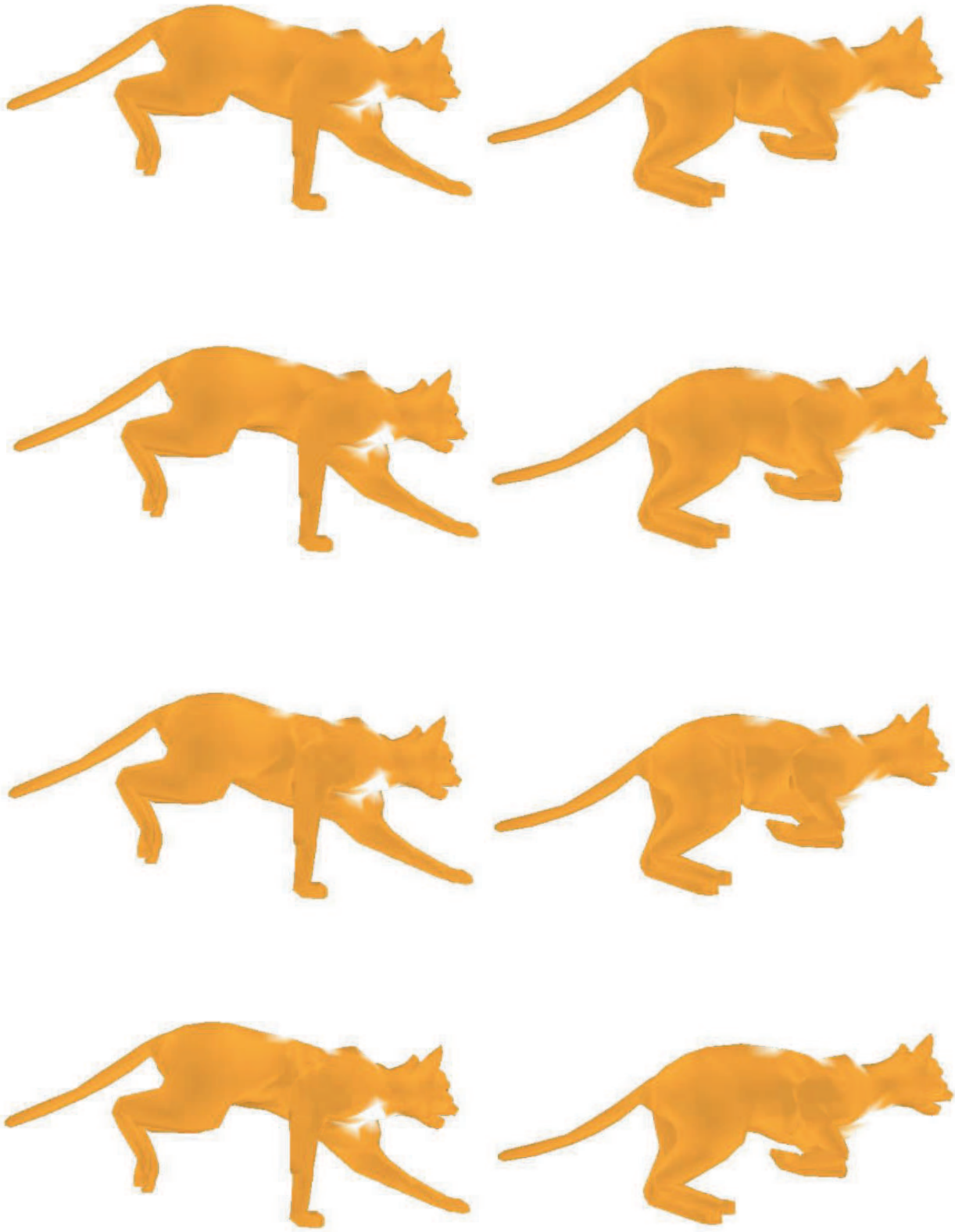


FIG. 6.5 – Mise en évidence de l'apport de nos outils. La première rangée montre des images extraites d'une animation par *skinning* classique. Sur la seconde rangée, nous avons ajouté notre *skinning dynamique* au niveau du ventre. La troisième rangée montre la même animation uniquement avec des séries de plis. Enfin, la denière rangée combine les deux outils.

FIG. 6.6 – Photo réelle d'un lynx en mouvement. Des petits plis se forment au niveau de la patte avant lorsque celui-ci se déplace (entourés d'une ellipse). Image extraite d'un documentaire animalier.

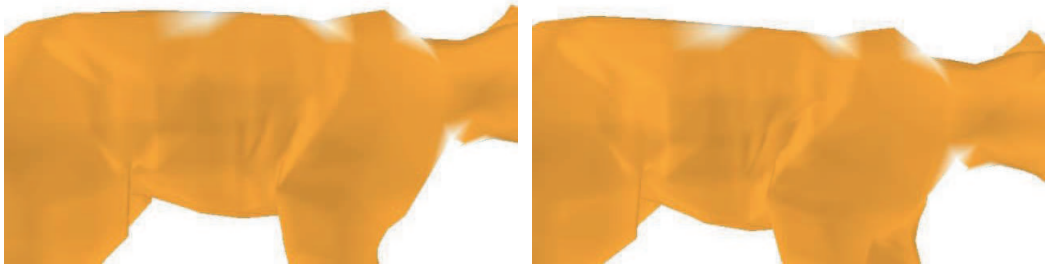
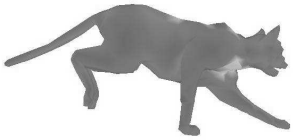


FIG. 6.7 – Deux extraits de l'animation de chat générée par *skinning* à laquelle nous avons appliqué notre *skinning dynamique* puis deux séries de plis.



CHAPITRE 7

Discussion sur l'intérêt et les limitations des méthodes géométriques

Mom ... Mom ! You're messing up my mane !

Simba, The Lion King

Ce chapitre a pour but de discuter nos choix principalement orientés vers l'utilisation de méthodes géométriques, couplées éventuellement à des éléments physiques minimalistes, par rapport à l'état de l'art qui regorge de simulations purement physiques ou anatomiques.

7.1 Comparaison sur un exemple test

Tout d'abord, afin d'effectuer une comparaison entre le *skinning dynamique* que nous avons introduit au chapitre 4 et les techniques physiques classiques, nous avons animé le modèle du champignon par une simulation physique de type masses-ressorts.

Pour effectuer notre simulation physique, nous avons tout d'abord dû calculer un maillage volumique de tétraèdres à partir de la surface polygonale du modèle. Pour cela, nous avons utilisé le logiciel TetMesh-GHS3D qui a l'avantage de conserver les triangles de la surface polygonale originale, ce qui facilite le rendu. Nous avons ensuite associé des masses aux nœuds des tétraèdres, et des ressorts aux arêtes. Afin de spécifier un *os*, nous avons sélectionné toutes les masses situées dans un cylindre positionné au centre du modèle, dans la direction du pied du champignon, pour leur infliger

une masse infinie et ainsi contrôler leur mouvement. Le mouvement de ces masses faisant partie de l'os permet ainsi de déplacer le modèle tout en amorçant les déformations dynamiques. Les résultats de la simulation sont montrés sur la figure 7.1, colonne (a).

La simulation physique a été implémentée sous OpenMASK[1], la plateforme de simulation développée au sein de l'équipe SIAMES et la méthode d'intégration utilisée est un schéma d'Euler implicite.

La première question qui se pose est celle de la qualité visuelle des résultats. Si l'on compare les résultats de la figure 7.1 (b), obtenus à l'aide de notre *skinning dynamique* surfacique, avec ceux de la figure 7.1 (a), obtenus par une simulation physique volumique, on se rend assez vite compte que les différences sont peu perceptibles. Les comportements que l'on peut générer sont à peu près les mêmes et ils sont obtenus en faisant varier le même type de paramètres (masse, raideur et amortissement d'un ressort).

Par contre, lorsque l'on s'intéresse aux gains en performance, les chiffres parlent d'eux mêmes : utilisation d'un seul ressort dans le cas du *skinning dynamique* contre environ 3000 dans le cas de la simulation masses-ressorts. En première approximation, on peut se dire que notre algorithme est 3000 fois moins coûteux. Mais ce n'est pas tout à fait vrai. Il faut aussi tenir compte du fait que nos ressorts (même dans le cas où il y en a plusieurs) ne sont pas interdépendants. De ce fait, les interactions entre ressorts sont nulles, ce qui simplifie l'intégration numérique d'une part, et permet d'utiliser des schémas d'intégration explicites puisque les risques de divergence sont grandement diminués. On se retrouve donc avec une simulation plus stable et moins coûteuse. Par ailleurs, le fait que les ressorts soient totalement indépendants peut permettre une parallélisation aisée de l'algorithme.

Le prix à payer est bien évidemment le manque de réalisme physique. En effet, notre algorithme ne simule que le mode de vibration principal, ce qui s'avère être une simplification importante. Cependant, pour pallier à ce manque de réalisme, il serait possible, comme nous l'avons détaillé au chapitre 4, de rajouter des ressorts supplémentaires, pour simuler des modes à plus haute fréquence. Cet ajout n'entraînerait pas un gros surcoût puisque les ressorts resteraient indépendants.

7.2 Contrôle de l'utilisateur

Que ce soit l'outil permettant de créer des plis ou bien le *skinning dynamique*, un avantage certain de nos techniques est la facilité de contrôle offerte à l'utilisateur. En effet, par rapport aux méthodes physiques, l'utilisateur ne se trouve pas face à des paramètres physiques incompréhensibles à régler, mais face à des paramètres qui ont un sens visuel. Si l'on prend par exemple le cas des plis générés par modèles physiques, l'utilisateur doit définir une raideur pour le tissu, un amortissement, des masses pour les nœuds du maillage. Dans notre technique, nous proposons des paramètres ayant un sens géométrique, tels que la forme générale des plis, leur largeur, leur hauteur, l'espacement des bosses.

L'autre avantage des méthodes géométriques sur les modèles physiques est que l'utilisateur à un contrôle local et direct. Que ce soit une série de plis ou un élément de chair dynamique, c'est l'utilisateur qui choisit à quel endroit ces effets doivent apparaître. Cette différence dans le contrôle est à rapprocher de celle qui existe entre les déformations directes des nœuds d'un maillage et les déformations par FFD.

[1] http ://www.openmask.org

(a) (b)

FIG. 7.1 – Comparaison de deux techniques de simulation différentes sur un même modèle de champignon. Colonne (a), le modèle est déformé à l'aide d'un réseau masses-ressorts $3D$ constitué d'environ 3000 ressorts amortis. Le volume est préalablement découpé en tétraèdres. Colonne (b), le modèle est animé par notre *skinning dynamique* à l'aide d'un seul ressort. Dans les deux cas, les déformations sont induites par le mouvement d'un squelette interne rigide et les résultats visuels obtenus sont similaires.

7.3 Simplicité de modélisation

Nos méthodes se basent sur un *skinning* existant. Elles ne nécessitent pas la création de muscles individuels comme les méthodes anatomiques. Ici, la forme cinématique de la déformation est apportée par le *skinning*. Si, il y a quelques années, les modèles anatomiques étaient indispensables au réalisme des déformations, ce n'est plus vrai aujourd'hui. Des déformations cinématiques de qualité peuvent être obtenues par *skinning*, qui est une méthode plus simple à mettre en place et temps-réel. Notre outil reste dans le même esprit, que ce soit le *skinning dynamique* qui propose des déformations à partir de poids ou bien les plis qui sont définis en quelques secondes par le tracé d'un segment de droite sur la surface à déformer. Les deux techniques sont donc très simples et rapides à utiliser.

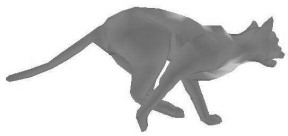
7.4 Facilité d'implémentation

Les deux algorithmes sont également simples à implémenter. À part la subdivision locale du maillage nécessaire pour l'utilisation des plis, toutes les autres étapes de l'algorithme manipulent les sommets du maillage, sans connaissance de leur voisinage. Ils sont donc adaptés à leur utilisation au sein d'un *vertex program* de carte graphique récente, ce qui serait intéressant pour appliquer ces techniques à un grand nombre de personnages en mouvement.

7.5 Validation des techniques

L'élaboration des deux algorithmes présentés dans cette thèse a été effectuée dans un souci d'apporter des outils directement utilisables par les infographistes. Nous avons donc essayé de combler les besoins de ces derniers en apportant des solutions à des problèmes spécifiques et en comparant les difficultés et temps d'utilisation de nos méthodes par un utilisateur non entraîné (moi même) par rapport à l'utilisation de MAYA par Christine Depraz, infographiste professeur aux Gobelins. Sur l'exemple de la création de plis, la comparaison est étonnante. Alors que l'infographiste a passé plusieurs dizaines de minutes pour mettre en place une série de formes clefs, avec notre technique, il n'a fallu qu'une ou deux minutes pour animer des plis équivalents.

Bien que n'ayant pas manipulé directement, Christine Depraz a pu juger de l'efficacité de nos outils en me regardant les manipuler. Elle a de plus souligné la simplicité d'utilisation et l'intérêt des techniques géométriques pour les artistes, qui ne peuvent pas toujours se permettre se passer beaucoup de temps pour l'ajout de détails aux animations, et qui sont en grande demande de tels outils.



CHAPITRE 8

Conclusion et travaux futurs

Fish are friends, not food.

Bruce, Finding Nemo

Nous avons montré, dans la première partie de ce document, que bien que la technique de *skinning* soit aujourd'hui la meilleure technique d'animation temps-réel existante, elle ne pouvait fournir que des déformations cinématiques de la peau de personnages, et avec peu de détails dynamiques comme la vibration des tissus ou les plis de la peau.

Nous avons donc proposé, aux chapitres 4 et 5, deux outils permettant d'ajouter ces détails en temps-réel à une animation existante.

Le premier outil permet d'ajouter des effets dynamiques à des éléments de chair définis par l'utilisateur, à l'aide d'un ressort amorti et d'un ensemble de poids décrivant la déformation. Les paramètres utilisateur sont intuitifs et peu nombreux à régler, et, la technique étant temps-réel, il est aisé d'ajuster ces paramètres pendant l'animation.

Notre second outil, en simulant la résistance de la peau à la compression, permet, à travers une méthode complètement géométrique, de créer des séries de plis dynamiques à la surface de la peau ou des vêtements. Là aussi, les paramètres sont très intuitifs puisqu'il s'agit, pour l'utilisateur, de choisir une forme des plis, leur espacement ou leur hauteur maximale.

Le chapitre 6 a montré quelques résultats illustrant l'utilisation de nos deux outils sur une même animation. Ces contributions apportent des solutions élégantes aux problèmes que nous nous étions posés. Elles ne ralentissent pas ou peu les performances temps-réel du *skinning* et sont adaptées pour une utilisation dans les jeux vidéos. Par rapport aux méthodes anatomiques ou physiques, nos outils ne sont pas difficiles à mettre en place, ils ne demandent pas d'adaptation particulière du maillage ou de la représentation de l'objet, et ils tirent pleinement profit de l'existant.

Néanmoins, des améliorations sont encore possibles, et ce, à plusieurs niveaux. Tout d'abord au niveau des paramètres et des différentes courbes d'atténuation utilisées. De nombreuses autres fonctions pourraient être explorées, à la fois pour diversifier les formes que l'on peut générer, et aussi pour favoriser la liberté de l'infographiste. À l'opposé, pour permettre à des utilisateurs non experts de créer des plis par exemple, il serait aussi intéressant de pouvoir proposer un réglage automatique des paramètres. Une piste prometteuse serait l'utilisation de vidéos ou de photos réelles d'un personnage, de son visage ou de ses vêtements par exemple, dans des expressions variées, pour en extraire automatiquement la courbe de contrôle, le profil des rides et leur largeur.

Par ailleurs, on retrouve deux pistes communes pour l'extension de nos algorithmes au niveau de leur implémentation. Étant destinées principalement au domaine du jeu vidéo, il serait très intéressant de programmer ces deux techniques sur une carte graphique récente. Les deux algorithmes ont été construits dans cet esprit et se prêteraient bien *a priori* à une adaptation sur *vertex program*, mis à part peut-être la subdivision locale du maillage pour les plis dynamiques. D'autre part, ces contributions étant en quelque sorte des outils proposés aux artistes pour créer des animations plus réalistes, nous aimerions intégrer ces algorithmes dans un logiciel du commerce tel que Maya [ali]. Cela nous permettrait de diffuser nos outils et de bénéficier d'un retour de la part des utilisateurs, afin d'identifier les problèmes éventuels des deux techniques et de les améliorer.

TABLE DES MATIÈRES

1 Introduction **5**

I État de l'art **9**

2 Modélisation d'objets déformables structurés et animation traditionnelle par squelette **13**

2.1 Représentation de l'enveloppe d'un objet 3D 13

2.1.1 Surfaces polygonales 14

2.1.2 Surfaces paramétriques 14

2.1.3 Surfaces de subdivision 18

2.1.4 Surfaces implicites 20

2.2 Animation des modèles structurés à squelettes 22

2.2.1 Squelette et articulations 23

2.2.2 Cinématique directe 23

2.2.3 Cinématique inverse 25

2.2.4 Animation dynamique 26

2.2.5 Capture du mouvement 26

2.3 Techniques de déformation 27

2.3.1 Techniques géométriques 27

2.3.1.1 Interpolation de formes clefs 27

2.3.1.2 Déformations dans les sous-espaces du squelette 28

2.3.1.3 Déformations de formes libres 30

2.3.2 Modèles physiques 31

2.3.2.1 Quelques éléments de physique 31

2.3.2.2 Les différents modèles 34

2.3.2.3 Intégration numérique 37

3 Modélisation et déformation des muscles, tissus et peau de personnages 39

3.1 Le point de vue anatomique 39
3.1.1 Définitions 39
3.1.2 Propriétés mécaniques 41
3.2 Habillage géométrique surfacique et ses déformations 42
3.2.1 Interpolation de formes clefs 42
3.2.2 Déformations dans les sous-espaces du squelette 44
3.2.3 Cas des modèles paramétriques 46
3.3 Modélisation des muscles pour un modèle en couches 50
3.3.1 Modèles géométriques 50
3.3.2 Modèles physiques 56
3.4 Tissus 60
3.4.1 Masses-ressorts 60
3.4.2 Éléments finis 61
3.5 Peau élastique 68
3.5.1 Surface élastique 69
3.5.2 Surface triangulée 69
3.6 Synthèse des détails externes : plis de la peau et/ou de vêtements 70
3.6.1 Bump et displacement mapping 71
3.6.2 Déformations géométriques du maillage 72

II Contributions 77

4 Skinning dynamique 81

4.1 Contraintes sur la déformation 81
4.1.1 Modèle mécanique 82
4.1.2 Localisation des déformations 84
4.1.3 Rôle des os 84
4.2 Principe général 85
4.3 Élément de chair dynamique 86
4.3.1 Spécification d'un élément dynamique 87
4.3.2 Repère dynamique 88
4.3.3 Application au maillage 90
4.4 Calcul automatique des poids de *skinning dynamique* 91
4.4.1 Morphologie de l'élément 92
4.4.2 Atténuation vers les extrémités 93
4.4.3 Normalisation 95
4.4.4 Élongation maximale du ressort 95

4.5 Limitations et solutions 96
4.6 Algorithme et performance 97
4.7 Quelques Résultats 99
4.7.1 Un cas simple : le champignon 99
4.7.2 Exemple de mouvement rapide sur un animal 100
4.7.3 Animation d'un humanoïde 100
4.8 Conclusion et perspectives 103

5 Plis dynamiques en temps-réel 107
5.1 Les plis dans la nature 108
5.1.1 Propagation dans la zone de compression 109
5.1.2 Atténuation 110
5.1.3 Échelles 110
5.2 Principe de la technique 111
5.3 Déformation de la courbe de contrôle 112
5.3.1 Courbe de contrôle 112
5.3.2 Algorithme de déformation 113
5.3.3 Conservation de la longueur 113
5.3.4 Différentes stratégies possibles 114
5.3.5 Niveaux de détails 115
5.4 Déformation du maillage 116
5.4.1 Mise en place de l'outil 116
5.4.2 Région d'influence 117
5.4.3 Atténuation aux bords de la région d'influence 117
5.4.4 Déplacement des sommets influencés 118
5.4.5 Subdivision du maillage 120
5.4.6 Plis superposés 122
5.5 Résultats et performances 122
5.5.1 Rides du front 122
5.5.2 Plis au niveau des articulations 123
5.5.3 Plis de vêtements 123
5.6 Limitations et extensions de l'algorithme 124
5.6.1 Courbe de contrôle multi-résolution 124
5.6.2 Rides courbes 125
5.7 Conclusion et travaux futurs 126

6 Combinaison des deux outils 129
6.1 Résultats 129
6.2 Comparaison avec un *skinning* simple 129
6.3 Comparaison avec une situation réelle 130

7 Discussion sur l'intérêt et les limitations des méthodes géométriques 135
7.1 Comparaison sur un exemple test 135
7.2 Contrôle de l'utilisateur 136
7.3 Simplicité de modélisation 138
7.4 Facilité d'implémentation 138
7.5 Validation des techniques 138

8 Conclusion et travaux futurs 139

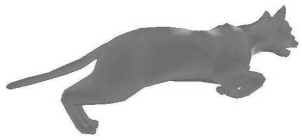


TABLE DES FIGURES

2.1 Un modèle de chat : (a)-le modèle affiché à l'écran ; (b)-affichage *fil de fer* qui montre les triangles le composant. 14

2.2 Différents exemples de courbes splines construites à partir des mêmes points de contrôle : (a)-spline d'interpolation ; (b)-B-spline ; (c)-spline cubique fermée ; (d)-B-spline cubique fermée. 16

2.3 Un exemple de surface $3D$ créée à partir de NURBS : (a)-les points de contrôle et les courbes NURBS définies par ces points ; (b)-la surface finale et l'influence locale d'un point de contrôle. 17

2.4 L'ajout d'un point de contrôle sur une surface paramétrique entraîne automatiquement l'ajout d'une colonne et d'une ligne entières de points de contrôle car il faut découper chaque domaine paramétrique. 18

2.5 Geri, personnage principal du film d'animation *Geri's Game*, a été modélisé par des surfaces de subdivision de type Catmull-Clark. 18

2.6 Un exemple didactique de subdivision par le schéma de Catmull-Clark : (a)-le maillage original composé de sommets v_i ; (b)-ajout de nouveaux sommets f_i au centre de chaque face ; (c)-ajout de nouveaux sommets e_i au centre de chaque arête et de nouveaux liens entre les nouveaux sommets f_i et e_i ; (d)-déplacement des anciens sommets pour obtenir le maillage final. 19

2.7 Une surface de subdivision et son processus de raffinement : (a)-maillage initial ; (b)-après une étape de subdivision ; (c)-après deux étapes ; (d)-après n étapes (images extraites de [HDD +94]). 19

2.8 Différents principes de subdivision existent : (a)-division de quadrilatères ; (b)-division de triangles ; (c)-division des sommets (images extraites de [ZSD +03]). 20

2.9 Arbre des actions possibles sur des primitives implicites pour obtenir un modèle détaillé (image extraite de [WGG99]). 21

2.10 Exemples de modèles créés à l'aide de surfaces implicites (images extraites de [FCG02, LAG01, HAC03]). 22

2.11 Un exemple de squelette sur le modèle du chat : (a)-le squelette seul et sa hiérarchie ; (b)-le squelette dans le modèle géométrique. 23
2.12 Articulations réelles et articulations idéales : (a)-lorsque deux membres d'une articulation réelle sont en mouvement relatif, les centrodes permettent de décrire les lieux des centres instantanés des deux objets et nous montrent le glissement qui intervient entre les deux membres (image extraite de [KG83]) ; (b)-en négligeant les translations, l'articulation peut être ramenée à une rotule (3 degrés de liberté) ou un modèle plus contraint (images extraites de [Mau99]). 24
2.13 Exemple de trajectoires d'un repère : 3 courbes décrivent les translations en x, y, et z et 3 courbes décrivent les rotations selon les mêmes axes. 24
2.14 Illustration de la technique du *skinning rigide* : le maillage est étiré d'un côté de l'articulation et il s'intersecte de l'autre côté (image extraite de [Blo02]). 28
2.15 Illustration de la technique du *skinning* utilisant deux repères : le point P se déplace en un nouveau point P qui se trouve obligatoirement sur la droite reliant P_{vert} à P_{rouge}. 29
2.16 Problèmes liés à la technique du *skinning* : (a)-aplatissement au niveau du coude ; (b)-amincissement au niveau de l'articulation quand le bras est vrillé. 30
2.17 Loi de comportement d'un matériau élastique qui trace l'évolution des déformations (en abscisse) en fonction des contraintes appliquées à l'objet (en ordonnée). 32

3.1 Musculature réelle. 40
3.2 Composition de la peau : (1)-épiderme ; (2)-derme, (3)-hypoderme. 41
3.3 Mise en valeur du gain des B-splines hiérarchiques par rapport aux B-splines classiques : (a)-surface plate avec une série de bosses sur la diagonale ; (b)-modèle présenté en (a) construit avec des B-splines hiérarchiques. 47
3.4 Un modèle de tête de dragon et ses niveaux de raffinement : (a)-les quatre premiers niveaux (le modèle de départ étant un simple tore) ; (b)-les quatre derniers niveaux avec un modèle de plus en plus détaillé. 47
3.5 Un exemple d'animation utilisant les B-splines hiérarchiques : (a)-modèle au repos ; (b)-modèle dans la pose courante de l'animation. 48
3.6 Deux exemples d'utilisation des FFDs surfaciques. Sur l'image de gauche, la chemise du personnage est déformée quand son bras bouge. Sur l'image de droite, on peut voir une déformation correcte de la zone du bassin du personnage et de sa jambe (images extraites de [SK00]). 49
3.7 Muscle proposé par Chadwick et al., modélisé à l'aide d'une paire de FFDs (image inspirée de [CHP89]). 51
3.8 Modèles de muscles de Scheepers et al. : (a)-multi-belly muscle model ; (b)-general muscle model (images extraites de [SPCM97]). 53
3.9 Illustration de la contraction sur le modèle de muscle de Wilhelms [Wil97]. On peut noter que le muscle est constitué de 3 ellipsoïdes : un central qui représente le muscle et un à chaque extrémité qui joue le rôle de tendon. 54
3.10 Muscles à base d'ellipsoïdes de Scheepers et al. [SPCM97]. 54
3.11 Modèle de muscle de Wilhelms, basé sur un cylindre déformé (image extraite de [WG97]). 55
3.12 Description du maillage masses-ressorts du muscle de Nedel (image inspirée de [Ned98]). 58
3.13 Plis réels sur l'avant (a) et l'arrière (b) d'un jean. 71

3.14 Résultats de rides du visage obtenus par Bando et al. en 2002 (images extraites de [BKN02]). 72
3.15 Résultats de plis obtenus par Combaz et Neyret en 2002 par une technique de simulation (images extraintes de [CN02]). 73

4.1 Une cuisse réelle : (a)-une radiographie : l'os, situé plus ou moins au centre, est entouré de muscles et de tissus ; (b)-représentation de l'enchevêtrement de muscles, tendons et tissus autour de l'os. 82
4.2 Un modèle mécanique simplifié d'un membre humain et ses déformations sous différentes accélérations : (a)-le modèle ; (b)-vue d'une coupe et labellisation de deux points du solide : P_1 et P_2 ; (c)-accélération de l'os dans la direction de son axe ; (d)-accélération de l'os dans la direction orthogonale à son axe ; (c) et (d)-P_1 et P_2 se trouvent déplacés dans la direction opposée à l'accélération, dP_1 et dP_2 décrivent ces déplacements. 83
4.3 Deux formes clefs : la forme dynamique est la translatée de la forme statique et son mouvement est déterminé par un ressort amorti attaché à l'os. 86
4.4 Un personnage articulé, un exemple de décomposition en *éléments de chair dynamique* et les systèmes de coordonnées locales correspondants. 87
4.5 Un squelette d'animation et sa hiérarchie de repères contenant un élément dynamique spécifié par deux repères : le repère dominant, plus haut dans la hiérarchie, et l'un de ses repères fils. 88
4.6 Un repère dynamique F_d est connecté au repère local de l'élément de chair dynamique F_s à l'aide d'un ressort amorti de longueur à vide nulle. (a) et (b) : le repère local se trouve au milieu de l'os virtuel délimité par un repère du squelette à chaque bout. (c) : le repère local est défini par 3 repères du squelette. 89
4.7 Comportement du ressort lors de l'accélération de l'os. 91
4.8 Le poids avant atténuation donné par la première contrainte correspond à la distance Euclidienne entre le sommet considéré et l'os virtuel de l'élément. Par exemple, P_1 aura comme poids d_1 et P_2, d_2. 92
4.9 La fonction d'atténuation diminue les poids obtenus par $shape(x, z)$ d'autant plus que sommet concerné est proche d'une des extrémités de l'élément. 93
4.10 Résultat de l'utilisation de différentes fonctions d'atténuation sur un élément de chair cylindrique ayant une épaisseur nulle de l'os : (a)-le cylindre non déformé ; (b)-l'utilisation d'un schéma linéaire non réaliste nous montre la nécessité de l'utilisation d'une fonction au minimum C1 ; (c)-la fonction parabole offre une déformation réaliste mais engendre des problèmes de continuité aux extrémités ; (d)-la fonction polynomiale de Wyvill décroît avec une tangente nulle aux extrémités pour des temps de calcul raisonnables. 94
4.11 Effets de la fonction d'atténuation : (a)-un exemple de shape(x) ; (b)-la fonction d'atténuation en forme de parabole (trait plein) et la fonction d'atténuation de Wyvill (en pointillés) ; (c)-la fonction des poids correspondante avant normalisation, pour chaque fonction d'atténuation présentée en (b). Observez la tangente non nulle présente sur la courbe des poids obtenue avec la parabole comme fonction d'atténuation (trait plein), alors que la courbe obtenue grâce à la fonction de Wyvill a des tangentes nulles, ce qui évite les discontinuités du maillage après l'étape de *skinning dynamique* (en pointillés). 95

4.12 Un exemple de fonction $shape$ non linéaire qui ne garantit pas que la chair ne traversera pas l'os : (a)-la fonction $shape$ (en trait plein) est linéaire et les courbes de poids avant normalisation (atténuation parabolique en petits pointillés, atténuation de Wyvill en pointillés larges) ne croisent jamais la courbe $shape$. L'élongation maximale correspondante peut garantir que la chair ne traversera jamais l'os ; (b)-la fonction $shape$ non linéaire pose des problèmes avec l'une des courbes d'atténuation (parabole) qui la *croise*. L'élongation maximale ne peut pas garantir à elle seule que la chair ne traversera pas l'os. 96

4.13 Un exemple d'élément dégénéré : (a)-l'élément dégénéré avec mise en évidence de la zone à problèmes ; (b)-ajout d'un repère au squelette pour résoudre le problème. . 97

4.14 Un modèle de champignon surfacique déformé par notre méthode de *skinning dynamique*. Le mouvement est induit par un squelette rigide placé au milieu du modèle et qui est défini par deux repères : le repère dominant au niveau du chapeau du champignon et son repère fils à la base du pied. Les déformations obtenues sont à comparer à celles de la figure 7.1 qui a été simulée par un réseau volumique de masses et de ressorts. Avec notre technique, un seul ressort amorti a été nécessaire. 99

4.15 Notre modèle de champignon de masse $39.35\ Kg$, soumis à une gravité variable : plus la gravité est forte, plus le modèle est déformé. 100

4.16 Animation d'un modèle de chat, en temps-réel, en utilisant notre méthode de $skinning\ dynamique$. La première image montre les poids associés aux sommets du maillage (ils varient de 0 en noir à 1 en blanc). La dernière image montre le ressort (le carré rose est le point fixe, le bleu est l'extrémité libre du ressort). 101

4.17 Le même modèle de chat, déformé en utilisant le même élément de chair dans la région du ventre, dans un mouvement de saut. 101

4.18 Animation de la poitrine d'un personnage : l'image de gauche montre l'ensemble des valeurs des poids (variant de noir pour 0 à blanc pour 1) ; l'image de droite montre le modèle non déformé dans sa position de repos. 102

4.19 Animation de la poitrine d'un personnage : visualisation des déformations en affichage *fil de fer* ainsi que du mouvement du ressort amorti. 102

4.20 Animation de la poitrine d'Eve par un élément de chair dynamique. 103

4.21 Le ventre, les fesses, les bras et la poitrine de Eve ont été animés en temps-réel par quatre éléments de chair connectés (utilisant donc quatres ressorts amortis). La première rangée d'images nous montre les poids de *skinning dynamique* associés aux ressorts ainsi que l'emplacement de ces derniers. La seconde rangée nous montre quelques images extraites de l'animation. 104

4.22 Prise en compte de la torsion du ressort : (a)-première possibilité, on ajoute un couple de torsion au ressort amorti (image extraite de [MPT99]) ; (b)-seconde possibilité, on utilise trois ressorts amortis au lieu d'un seul. 105

5.1 Des exemples de plis : (a)-un gros pli statique sur un personnage de synthèse ; (b)-des plis dynamiques réels. 108

5.2 Des plis dans la nature : de gauche à droite et de haut en bas, on peut voir un avant bras, le dos d'une main, un poignet et un t-shirt. 109

5.3 Apparition de plis simultanément tout le long de la zone de compression. Les plis se forment orthogonalement à la direction de compression. 110

5.4 Apparition de plis auprès des zones d'application de forces puis propagation vers le milieu de la zone de compression. Les plis se forment orthogonalement à la direction de compression. 110

5.5 Apparition de plis auprès de la zone d'application de la force puis propagation vers la zone non déformée. Les plis se forment orthogonalement à la direction de compression : (a)-sur de la peau ; (b)-sur du tissu. 111

5.6 Un gros pli réel : sa hauteur est maximale au centre et s'atténue vers les extrémités (image reproduite avec l'accord de Zabador). 111

5.7 Une main réelle montrant plusieurs rides à différents niveaux de détails. 112

5.8 Une courbe de contrôle et ses déformations quand l'extrémité A se rapproche de l'extrémité B, fixe. Les bosses apparaissent d'abord auprès de A puis se propagent. 113

5.9 Système de coordonnées $2D$ associé à la courbe de contrôle. L'axe des abscisses est porté par le segment $\vec{AB}$ et l'axe des ordonnées lui est orthogonal. 114

5.10 Le rétrécissement horizontal d est réinjecté en déplaçant P_1 dans la direction verticale d'une hauteur h. 114

5.11 Différents schémas de courbes de contrôle : 1.-l'espacement entre les bosses est de 1 segment, la crête de chaque bosse est composée d'un seul point de contrôle ; 2.-l'espacement entre les bosses est de 2 segments ; 3.-la crête de chaque bosse est composée de 2 points de contrôle. 115

5.12 Rides à différents niveaux de détails : (a)-une courbe de contrôle subdivisée ; (b)-application de cette courbe sur un maillage. 116

5.13 Plan de déformation de la courbe. 117

5.14 Une région d'influence est associée à chaque courbe de contrôle : (a)-la région d'influence et son système de coordonnées associé ; (b)-un cas simulé. 118

5.15 Un profil d'atténuation définit comment les bosses décroissent jusqu'à s'annuler aux bords du rectangle d'influence ; (a) $y = 1 - abs(x)$; (b) $y = 1 + \frac{(-4x^6+17x^4-22x^2)}{9}$. . . . 119

5.16 Mise en évidence de la nécessité d'utiliser une fonction d'atténuation continue C^1 : (a)-pas d'atténuation ; (b)-atténuation linéaire ; (c)- atténuation non linéaire utilisant la fonction de Wyvill. 119

5.17 Calcul de l'élévation d'un sommet influencé : l'abscisse de la projection du point P dans le rectangle d'influence donne la hauteur h qui est atténuée par a calculée à partir de son ordonnée. 120

5.18 Importance du rapport entre le *pas* de la courbe de contrôle et la résolution du maillage : (a)-la courbe de contrôle est beaucoup trop échantillonnée par rapport au maillage, une seule grosse bosse apparaît au lieu d'une série de rides très fines ; (b)-le maillage est beaucoup plus fin que le *pas* de la courbe, les plis $3D$ reflètent correctement la déformation de la courbe de contrôle. 121

5.19 Un exemple de maillage subdivisé localement à un niveau 3. On peut remarquer les triangles très étirés au bord de la zone subdivisée. 121

5.20 Superposition de plusieurs séries de rides sur l'arrière du genou d'un pantalon virtuel. Lorsque le genou se plie, les rides apparaissent. 122

5.21 Utilisation de notre outil pour créer des rides sur les fronts de deux modèles différents : un modèle d'homme pour la rangée du haut et de femme pour la rangée du bas. (a) et (d)-maillages non déformés ; (b), (c), (e) et (f)-maillages déformés avec différents paramètres. 123

5.22 Illustration de l'utilisation de notre outil sur un poignet de bébé. (a) et (b)-animation standard n'utilisant que le *skinning* classique : pas de rides présentes ; (c) et (d)-notre outil fait apparaître des plis dans le creux du poignet pendant le mouvement. . 124

5.23 Illustration de notre outil de création de plis sur un pantalon virtuel. La ligne du haut montre l'arrière d'un pantalon où les plis sont créés par deux séries de plis superposées. La ligne du bas montre l'avant d'un pantalon où une série de rides est disposée sur la jambe de droite. On peut constater la différence avec la jambe de gauche, dépourvue de plis. 125

6.1 Ajout de deux séries de plis sur le modèle de chat : une série entre la patte avant et le ventre, et une seconde série entre la patte arrière et le ventre. 130

6.2 Lorsque le chat se déplace, des plis se forment au niveau de la patte avant. Cette image montre une vue rapprochée de dessous de cette zone. 131

6.3 Animation du modèle du chat sur laquelle des effets dynamiques et des plis dynamiques ont été ajoutés. 131

6.4 Même animation que sur la figure 6.3, affichage en fil de fer. Noter le mouvement du ressort dynamique ainsi que le raffinement du maillage à la volée, uniquement lorsque les plis dynamiques sont actifs. 132

6.5 Mise en évidence de l'apport de nos outils. La première rangée montre des images extraites d'une animation par *skinning* classique. Sur la seconde rangée, nous avons ajouté notre *skinning dynamique* au niveau du ventre. La troisième rangée montre la même animation uniquement avec des séries de plis. Enfin, la denière rangée combine les deux outils. 133

6.6 Photo réelle d'un lynx en mouvement. Des petits plis se forment au niveau de la patte avant lorsque celui-ci se déplace (entourés d'une ellipse). Image extraite d'un documentaire animalier. 134

6.7 Deux extraits de l'animation de chat générée par *skinning* à laquelle nous avons appliqué notre *skinning dynamique* puis deux séries de plis. 134

7.1 Comparaison de deux techniques de simulation différentes sur un même modèle de champignon. Colonne (a), le modèle est déformé à l'aide d'un réseau masses-ressorts $3D$ constitué d'environ 3000 ressorts amortis. Le volume est préalablement découpé en tétraèdres. Colonne (b), le modèle est animé par notre *skinning dynamique* à l'aide d'un seul ressort. Dans les deux cas, les déformations sont induites par le mouvement d'un squelette interne rigide et les résultats visuels obtenus sont similaires. 137

BIBLIOGRAPHIE

[3ds] Discreet, 3ds Max. – http ://www.discreet.com/3dsmax/.

[ACP02] Brett Allen, Brian Curless et Zoran Popović. – Articulated body deformation from range scan data. In *Proceedings of SIGGRAPH'02, ACM Transactions On Graphics*, pp. 612–619. – Juillet 2002.

[AG01] Samir Akkouche et Eric Galin. – Adaptive implicit surface polygonization using marching triangles. *Computer Graphics Forum*, vol. 20, n° 2, pp. 67–80, 2001.

[ali] Alias-Wavefront, Maya. – http ://www.aliaswavefront.com.

[Arn94] Bruno Arnaldi. – Modèles physiques pour l'animation. – Habilitation à diriger des recherches, Université de Rennes 1, Février 1994.

[Aub02] Amaury Aubel. – *Anatomically-Based Human Body Deformations*. – Thèse de Doctorat, EPFL, Lausanne, Mai 2002.

[Bae01] Paolo Baerlocher. – *Inverse Kinematics Techniques for the Interactive Posture Control of Articulated Figures*. – Thèse de Doctorat, Ecole Polytechnique Fédérale de Lausanne, 2001.

[BKK +85] N.I. Badler, J.D. Korein, J.U. Korein, G.M. Radack et L.S. Brotman. – Positioning and animating human figures in a task-oriented environment. *The Visual Computer*, vol. 1, n° 4, pp. 212–220, Décembre 1985.

[BKMTK00] Laurence Boissieux, Gergo Kiss, Nadia Magnenat-Thalmann et Prem Kalra. – Simulation of skin aging and wrinkles with cosmetics insight. In *Computer Animation and Simulation 2000*, pp. 15–27. – Aoũt 2000.

[BKN02] Yosuke Bando, Takaaki Kuratate et Tomoyuki Nishita. – A simple method for modeling wrinkles on human skin. In *Proceedings of Pacific Graphics*. – Octobre 2002.

[BLCD02] Christoph Bregler, Lorie Loeb, Erika Chuang et Hrishi Deshpande. – Turning to the masters : Motion capturing cartoons. In *Proceedings of SIGGRAPH'02, ACM Transactions on Graphics*, pp. 399–407. – Juillet 2002.

[Bli78] James F. Blinn. – Simulation of wrinkled surfaces. In *Proceedings of SIGGRAPH'78*. pp. 286–292. – ACM Press, 1978.

[Bli82] J. Blinn. – A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, pp. 235–256, Juillet 1982.

[Blo87] Jules Bloomenthal. – *Polygonization of Implicit Surfaces*. – Rapport technique n° CSL-87-2, Xerox, Mai 1987.

[Blo02] Jules Bloomenthal. – Medial-based vertex deformation. In *Proceedings of the ACM SIGGRAPH Symposium on Computer Animation*. pp. 147–151. – ACM Press, 2002.

[BMW87] N.I. Badler, K.H. Manoochehri et G. Walters. – Articulated figure positioning by multiple constraints. *IEEE Computer Graphics and Applications*, vol. 7, n° 6, pp. 28–38, Juin 1987.

[BN95] Morten Bro-Nielsen. – Modelling elasticity in solids using active cubes - application to simulated operations. In *Proceedings of the First International Conference on Computer Vision, Virtual Reality and Robotics in Medicine*. pp. 535–541. – Springer-Verlag, Avril 1995.

[BS04] Jeff Bolz et Peter Schroder. – Evaluation of subdivision surfaces on programmable graphics hardware. – submitted for publication 2004.

[BW90] Jules Bloomenthal et Brian Wyvill. – Interactive techniques for implicit modeling. *Computer Graphics*, vol. 24, n° 2, pp. 109–116, Mars 1990.

[BW98] David Baraff et Andrew Witkin. – Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. pp. 43–54. – ACM Press, 1998.

[BWK03] David Baraff, Andrew Witkin et Michael Kass. – Untangling cloth. In *Proceedings of SIGGRAPH'03, ACM Transactions on Graphics*, pp. 862–870. – Juillet 2003.

[Can90] Marie-Paule Cani. – *Déformation de surfaces complexes : techniques de haut niveau pour la modélisation et l'animation*. – Thèse de Doctorat, Université Paris XI, Octobre 1990.

[Can93] Marie-Paule Cani. – An implicit formulation for precise contact modeling between flexible solids. In *Computer Graphics*. ACM, pp. 313–320. – Aoũt 1993. Published under the name Marie-Paule Gascuel.

[CB04] Benoît Le Callennec et Ronan Boulic. – Interactive motion deformation with prioritized constraints. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. pp. 163–171. – ACM SIGGRAPH / Eurographics, Aoũt 2004.

[CC78] Ed Catmull et Jim Clark. – Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, vol. 10, n° 6, pp. 350–355, Octobre 1978.

[CDA98] Stéphane Cotin, Hervé Delingette et Nicholas Ayache. – *Real-time elastic deformations of soft tissues for surgery simulation*. – Rapport technique n° 3511, Projet Epidaure, INRIA, Octobre 1998.

[CGC +02a] Steve Capell, Seth Green, Brian Curless, Tom Duchamp et Zoran Popovi ? – A multiresolution framework for dynamic deformations. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation*. pp. 41–47. – ACM Press, 2002.

[CGC +02b] Steve Capell, Seth Green, Brian Curless, Tom Duchamp et Zoran Popovic. – Interactive skeleton-driven dynamic deformations. In *Proceedings of SIGGRAPH'02, ACM Transactions on Graphics*, pp. 586–593. – Juillet 2002.

[CHP89] John E. Chadwick, David R. Haumann et Richard E. Parent. – Layered construction for deformable animated characters. In *Proceedings of SIGGRAPH'89, ACM Computer Graphics*. pp. 243–252. – ACM Press, 1989.

[CJ91] Sabine Coquillart et Pierre Jancéne. – Animated free-form deformation : an interactive animation technique. In *Proceedings of SIGGRAPH'91*. pp. 23–26. – ACM Press, 1991.

[CK02] K. Choi et H. Ko. – Stable but responsive cloth. *Proceedings of SIGGRAPH'02, ACM Transactions on Graphics*, vol. 21, n° 3, pp. 604–611, Juillet 2002.

[CN02] Jean Combaz et Fabrice Neyret. – Painting folds using expansion textures. In *Proceedings of Pacific Graphics'02*. – Octobre 2002.

[Coq90] Sabine Coquillart. – Extended free-form deformation : a sculpturing tool for 3d geometric modeling. In *Proceedings of SIGGRAPH'90*. pp. 187–196. – ACM Press, 1990.

[CZ92] David T. Chen et David Zeltzer. – Pump it up : Computer animation of a biomechanically based model of muscle using the finite element method. *Computer Graphics*, vol. 26, n° 2, Juillet 1992.

[DDCB01] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani et Alan H. Barr. – Dynamic real-time deformations using space and time adaptive sampling. In *Proceedings of SIGGRAPH'01*. – ACM Press, Aoũt 2001.

[Deb00] Gilles Debunne. – *Animation multirésolution d'objets déformables en temps-réel : application à la simulation chirurchicale*. – Thèse de Doctorat, Institut National Polytechnique de Grenoble, Décembre 2000.

[dJB04] Christina deJuan et Bobby Bodenheimer. – Cartoon textures. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. pp. 267–276. – ACM SIGGRAPH / Eurographics, Aoũt 2004.

[DLG90] Nira Dyn, David Levine et John A. Gregory. – A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, vol. 9, n° 2, pp. 160–169, Avril 1990.

[DTC96] Mathieu Desbrun, Nicolas Tsingos et Marie-Paule Cani. – Adaptive sampling of implicit surfaces for interactive modelling and animation. *Computer Graphics Forum*, vol. 15, n° 5, pp. 319–325, Décembre 1996.

[FB88] David R. Forsey et Richard H. Bartels. – Hierarchical b-spline refinement. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. pp. 205–212. – ACM Press, 1988.

[FCG02] Eric Ferley, Marie-Paule Cani et Jean-Dominique Gascuel. – Resolution adaptive volume sculpting. *Graphical Models (GMOD)*, vol. 63, pp. 459–478, Mars 2002.

[For91] David R. Forsey. – A surface model for skeleton-based character animation. *Second Eurographics Workshop on Animation and Simulation, Vienna, Austria*, pp. 55–73, Septembre 1991.

[FRDC04] Laurent Favreau, Lionel Reveret, Christine Depraz et Marie-Paule Cani. – Animal gaits from video. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. pp. 277–286. – ACM SIGGRAPH / Eurographics, Aoũt 2004.

[Fun93] Y.C. Fung. – *Biomechanics - Mechanical Properties of Living Tissues*. – Springer-Verlag, 1993, second édition.

[FvdPT97] Petros Faloutsos, Michiel van dePanne et Demetri Terzopoulos. – Dynamic free-form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, n° 3, pp. 201–214, 1997.

[Gle98] Michael Gleicher. – Retargetting motion to new characters. In *Proceedings of SIGGRAPH'98*, pp. 33–42. – 1998.

[GM85] M. Girard et A.A. Maciejewski. – Computational modeling for the computer animation of legged figures. *Computer Graphics*, vol. 19, n° 3, pp. 263–270, Juillet 1985.

[GP89] J. Griessmair et W. Purgathofer. – Deformation of solids with trivariate b-splines. In *Proceedings of Eurographics'89*, éd. par F.R.A Hopgood Hanmann, W. et W. Strasser. pp. 137–148. – Elsevier Science Publishers (North Holland), 1989.

[GTT89] Jean-Paul Gourret, Nadia Magnenat Thalmann et Daniel Thalmann. – Simulation of object and human skin deformations in a grasping task. In *Proceedings of SIGGRAPH'89, ACM Computer Graphics*. pp. 21–30. – ACM Press, 1989.

[HAC03] Samuel Hornus, Alexis Angelidis et Marie-Paule Cani. – Implicit modelling using subdivision-curves. *The Visual Computer*, vol. 19, n° 2-3, pp. 94–104, May 2003.

[HBVMT99] Sunil Hadap, Endre Bangerter, Pascal Volino et Nadia Magnenat-Thalmann. – Animating wrinkles on clothes. In *IEEE Visualization '99*. pp. 175–182. – IEEE Computer Society Press, Octobre 1999.

[HDD +94] Hugues Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John McDonald, Jean Schweitzer et Werner Stuetzle. – Piecewise smooth surface reconstruction. In *Proceedings of SIGGRAPH'94, Computer Graphics*, pp. 295–302. – Juillet 1994.

[Hod96] J.K. Hodgins. – Three-dimentional human running. In *Proceedings of the IEEE Conference on Robotics and Automation*. – 1996.

[HSIW96] A. Hilton, A. J. Stoddart, J. Illingworth et T. Windeatt. – Marching triangles : Range image fusion for complex object modelling. In *Proceedings of International Conference on Image Processing*. – 1996.

[HWBO95] J.K. Hodgins, W.L. Wooten, D.C. Brogan et J.F. O'Brien. – Animating human athletics. In *Proceedings of SIGGRAPH'95*. – 1995.

[JP02] Doug L. James et Dinesh K. Pai. – Dyrt : Dynamic response textures for real time deformation simulation with graphics hardware. In *Proceedings of SIGGRAPH'02, ACM Transactions on Graphics*, pp. 582–585. – San Antonio, TX, Juillet 2002.

[KG83] G. L. Kinzel et L. J. Gutkowski. – Joint models, degrees of freedom, and anatomical motion measurement. *Journal of Biomechanical Engineering*, vol. 105, pp. 55–61, 1983.

[KG03] Hiroshi Kono et Etsuo Genda. – Wrinkle generation model for 3d facial expression. *Sketches and Applications, SIGGRAPH'03*, Août 2003.

[KG04] Lucas Kovar et Michael Gleicher. – Automated extraction and parameterization of motions in large data sets. *ACM Transactions on Graphics*, vol. 23, n° 3, 2004.

[KGP02] Lucas Kovar, Michael Gleicher et Frederic Pighin. – Motion graphs. *ACM Transactions on Graphics*, vol. 21, n° 3, pp. 473–482, Juillet 2002.

[KJP02] Paul G. Kry, Doug L. James et Dinesh K. Pai. – Eigenskin : real time large deformation character skinning in hardware. In *Proceedings of the ACM SIGGRAPH Symposium on Computer animation*. pp. 153–159. – ACM Press, 2002.

[KM04] Tsuneya Kurihara et Natsuki Miyata. – Modeling deformable human hands from medical images. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. pp. 357–365. – ACM SIGGRAPH / Eurographics, Août 2004.

[Kob96] Leif Kobbelt. – Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Computer Graphics Forum*, vol. 15, n° 3, pp. 409–420, 1996.

[KZ04] Jan Klein et Gabriel Zachmann. – Point cloud collision detection. In *Proceedings of Eurographics'04*. – Septembre 2004.

[LAG01] Antoine Leclercq, Samir Akkouche et Eric Galin. – Mixing triangle meshes and implicit surfaces in character animation. In *Proceedings of Computer Animation and Simulation*. – Septembre 2001.

[LC87] William E. Lorensen et Harvey E. Cline. – Marching cubes : A high resolution 3d surface construction algorithm. In *Proceedings of SIGGRAPH'87, ACM Computer Graphics*. pp. 163–169. – ACM Press, 1987.

[LC04] Caroline Larboulette et Marie-Paule Cani. – Real-time dynamic wrinkles. In *Proceedings of Computer Graphics International'04*. pp. 522–525. – IEEE Computer Society Press, Juin 2004.

[LCA04] Caroline Larboulette, Marie-Paule Cani et Bruno Arnaldi. – Ajout de détails à une animation de personnage existante : cas des plis dynamiques. In *17e journées de l'AFIG*. – Novembre 2004.

[LCF00] J. P. Lewis, Matt Cordner et Nickson Fong. – Pose space deformation : A unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of SIGGRAPH'00, ACM Computer Graphics*. pp. 165–172. – ACM Press/Addison-Wesley Publishing Co., Juillet 2000.

[LCJ93] Francis Lazarus, Sabine Coquillart et Pierre Jancène. – *Interactive Axial Deformations*. – Rapport technique n° 1891, Projet Syntim, INRIA, Avril 1993.

[LMH00] A. Lee, H. Moreton et H. Hoppe. – Displaced subdivision surfaces. In *Proceedings of SIGGRAPH'00, ACM Computer Graphics*, pp. 85–94. – Juillet 2000.

[LN98] Thierry Lauthelier et Marc Neveu. – Mélange de formes pour l'animation de personnages. application en production. In *6èmes journées du groupe de travail animation, simulation, systèmes dynamiques*, pp. 38–51. – Octobre 1998.

[Loo87] C. Loop. – Smooth subdivision surfaces based on triangles. – Thèse de Master, University of Utah, 1987.

[Mau99] Walter Maurel. – *3D Modeling of the Human Upper Limb Including the Biomechanics of Joints, Muscles and Soft Tissues*. – Thèse de Doctorat, Ecole Polytechnique Fédérale de Lausanne, 1999.

[MBBT00] J.-S. Monzani, P. Baerlocher, R. Boulic et D. Thalmann. – Usign an intermediate skeleton and inverse kinematics for motion retargeting. In *Proceedings of Eurographics'00*. – 2000.

[MDM +02] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow et Barbara Cutler. – Stable real-time deformations. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation*. pp. 49–54. – ACM Press, 2002.

[MG03] Alex Mohr et Michael Gleicher. – Building efficient, accurate character skins from examples. In *Proceedings of SIGGRAPH'03, ACM Transactions on Graphics*. pp. 562–568. – ACM Press, 2003.

[MJ96] Ron MacCracken et Kenneth I. Joy. – Free-form deformations with lattices of arbitrary topology. In *Proceedings of SIGGRAPH '96*. pp. 181–188. – ACM Press, 1996.

[MK85] A.A. Maciejewski et C.A. Klein. – Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. *The International Journal of Robotics Research*, vol. 4, n° 3, pp. 109–117, 1985.

[MPT99] William A. McNeely, Kevin D. Puterbaugh et James J. Troy. – Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of SIGGRAPH'99*. pp. 401–408. – ACM Press/Addison-Wesley Publishing Co., 1999.

[Ned98] Luciana Porcher Nedel. – *Anatomic Modeling of Human Bodies Using Physically-Based Muscle Simulation*. – Thèse de Doctorat, EPFL Lausanne, 1998.

[NHK +85] T. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa et K. Omura. – Object modeling by distribution function and a method of image generation. *Journal of Papers at Electronics Communication Conference'85*, vol. J68-D, n° 4, pp. 718–725, 1985.

[Noc01] Olivier Nocent. – *Animation dynamique de corps déformables continus*. – Thèse de Doctorat, Université de Reims Champagne-Ardenne, Décembre 2001.

[NT98] Luciana Porcher Nedel et Daniel Thalmann. – Real time muscle deformations using mass-spring systems. In *Proceedings of Computer Graphics International'98*. pp. 156–165. – IEEE Computer Society Press, Juin 1998.

[NT00] Luciana Porcher Nedel et Daniel Thalmann. – Anatomic modeling of deformable human bodies. *The Visual Computer*, pp. 306–321, 2000.

[Pau81] R.P. Paul. – Robot manipulators : Mathematics, programming and control. *MIT Series in Artificial Intelligence*, 1981.

[PP01] Stefano Pasquariello et Catherine Pelachaud. – Greta : A simple facial animation engine. In *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*. – Septembre 2001.

[PW89] Alex Pentland et John Williams. – Good vibrations : Modal dynamics for graphics and animation. In *Proceedings of SIGGRAPH'89, ACM Computer Graphics*. pp. 207–214. – ACM Press, 1989.

[RNG99] Y. Remion, J.M. Nourrit et D. Gillard. – Dynamic animation of spline like objects. In *Proceedings of WSCG'99*, pp. 426–432. – 1999.

[RNN00] Yannick Rémion, Jean-Michel Nourrit et Olivier Nocent. – Dynamic animation of n-dimensional deformable objects. In *Proceedings of WSCG'00*. – 2000.

[SHB04] Basile Sauvage, Stefanie Hahmann et Georges-Pierre Bonneau. – Length preserving multiresolution editing of curves. *Computing*, vol. 72, pp. 160–171, 2004.

[SIC01] Peter-Pike J. Sloan, Charles F. Rose III et Michael F. Cohen. – Shape by example. In *Proceedings of the Symposium on Interactive 3D Graphics*, pp. 135–143. – Mars 2001.

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

[SM01] H.C. Sun et D.N. Metaxas. – Automating gait generation. In *Proceedings of SIGGRAPH'01*. – ACM Press / ACM SIGGRAPH, Aoũt 2001.

[SP86] Thomas W. Sederberg et Scott R. Parry. – Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH'86, ACM Computer Graphics*, pp. 151–160. – Aoũt 1986.

[SPCM97] Ferdi Scheepers, Richard E. Parent, Wayne E. Carlson et Stephen F. May. – Anatomy-based modeling of the human musculature. In *Proceedings of SIGGRAPH'97, ACM Computer Graphics*. pp. 163–172. – ACM Press/Addison-Wesley Publishing Co., 1997.

[SS88] L. Sciavicco et B. Siciliano. – A solution algorithm to the inverse kinematic problem for redundant manipulators. *IEEE Journal of Robotics and Automation*, vol. 4, pp. 403–410, 1988.

[TGB00] D. Tolani, A. Goswami et N.I. Badler. – Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models*, vol. 62, n° 5, pp. 353–388, Septembre 2000.

[TPBF87] Demetri Terzopoulos, John Platt, Alan Barr et Kurt Fleischer. – Elastically deformable models. In *Proceedings of SIGGRAPH'87, ACM Computer Graphics*, pp. 205–214. – Juillet 1987.

[TT93] Russell Turner et Daniel Thalmann. – The elastic surface layer model for animated character construction. *Proceedings of Computer Graphics International'93*, pp. 399–412, 1993.

[VMT99] Pascal Volino et Nadia Magnenat-Thalmann. – Fast geometrical wrinkles on animated surfaces. In *Seventh International Conference in Central Europe on Computer Graphics and Visualization (Winter School on Computer Graphics)*. – Février 1999.

[VY92] Marie-Luce Viaud et Hussein Yahia. – Facial animation with wrinkles. In *Eurographics Workshop on Animation and Simulation*. – Septembre 1992.

[WE84] W.A. Wolovich et H. Elliot. – A computational technique for inverse kinematics. In *Proceedings of the 32nd Conference on Decision and Control*, pp. 1359–1362. – Décembre 1984.

[WG97] Jane Wilhelms et Allen Van Gelder. – Anatomically based modeling. In *Proceedings of SIGGRAPH'97, ACM Computer graphics*. pp. 173–180. – ACM Press/Addison-Wesley Publishing Co., Aoũt 1997.

[WGG99] Brian Wyvill, Eric Galin et Andrew Guy. – Extending the csg tree. warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum*, vol. 18, n° 2, pp. 149–158, Juin 1999.

[WH94] A. Witkin et P. Heckbert. – Using particles to sample and control implicit surfaces. In *Proceedings of SIGGRAPH'94, Computer Graphics*, pp. 269–278. – Juillet 1994.

[Wil97] Jane Wilhelms. – Animals with anatomy. *IEEE Computer Graphics and Applications*, vol. 17, n° 3, pp. 22–30, Mai 1997.

[WKMMT99] Yin Wu, Prem Kalra, Laurent Moccozet et Nadia Magnenat-Thalmann. – Simulating wrinkles and skin aging. *The Visual Computer*, vol. 15, n° 4, pp. 183–198, 1999.

[WKT96] Yin Wu, Prem Kalra et Nadia Magnenat Thalmann. – Simulation of static and dynamic wrinkles of skin. In *Proceedings of Computer Animation '96*. pp. 90–97. – IEEE Computer Society Press, Juin 1996.

[WKT97] Yin Wu, Prem Kalra et Nadia Magnenat Thalmann. – Physically-based wrinkle simulation and skin rendering. In *Proceedings of EGCAS'97*. – 1997.

[WMW86] Geoff Wyvill, Craig McPheeters et Brian Wyvill. – Data structure for soft objects. *The Visual Computer*, vol. 2, n° 4, pp. 227–234, Aoũt 1986.

[WP02] Xiaohuan Corina Wang et Cary Phillips. – Multi-weight enveloping : least-squares approximation techniques for skin animation. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation*. pp. 129–138. – ACM Press, 2002.

[WTT94] Yin Wu, Nadia Magnenat Thalmann et Daniel Thalmann. – A plastic-visco-elastic model for wrinkles in facial animation and skin aging. In *Proceedings of Pacific Graphics'94*, pp. 201–213. – 1994.

[YN03] K. Yamane et Y. Nakamura. – Natural motion animation through constraining and deconstraining at will. *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, n° 3, pp. 352–360, Juillet 2003.

[ZB94] J. Zhao et N.I. Badler. – Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, vol. 13, n° 4, pp. 313–336, Octobre 1994.

[ZSD +03] Denis Zorin, Peter Schröder, Tony DeRose, Leif Kobbelt, Adi Levin et Win Sweldens. – Subdivision for modeling and animation. *Course Notes #23, SIGGRAPH'00*, Juillet 2003.

[ZSS96] Denis Zorin, Peter Schröder et Wim Sweldens. – Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH '96*. pp. 189–192. – ACM Press New York, NY, USA, 1996.

INSA de Rennes
Service des formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse : Traitement temps-réel des déformations de la peau et des tissus sous-cutanés pour l'animation de personnages.

Nom Prénom de l'auteur : LARBOULETTE Caroline

Membres du jury :

Monsieur	AKKOUCHE
Monsieur	REMION
Madame	CANI
Monsieur	CAMILLERAPP
Monsieur	NEVEU
Monsieur	PAZAT

Président du jury : Monsieur PAZAT

Date de la soutenance : 30/11/2004

Reproduction de la thèse soutenue :

- [x] Thèse pouvant être reproduite en l'état
- [ ] Thèse ne pouvant être reproduite
- [ ] Thèse pouvant être reproduite après corrections suggérées

Le Directeur,

P. FLEISCHMANN

Rennes, le 30/11/2004

Signature du Président du jury

Résumé

L'animation traditionnelle de personnages repose sur la création d'une enveloppe externe, appelée la peau, et d'un squelette interne constitué d'une hiérarchie de repères permettant de la déformer. De par son efficacité, le *skinning* est la technique de prédilection des infographistes. Cependant, bien que de nombreux travaux aient permis de résoudre les problèmes notoires de la méthode, certains types de déformations ne peuvent toujours pas être générés. On y retrouve les plis dynamiques de la peau ou des vêtements au niveau des articulations ou bien les effets d'inertie des masses musculaires et graisseuses lors de mouvements secs et rapides. Dans cette thèse, nous proposons deux techniques de modélisation et d'animation traitant ces problèmes spécifiques qui peuvent être utilisées sur une animation existante pour en accroître le réalisme.

Notre première contribution s'attache à ajouter, en temps-réel, des déformations dynamiques de la couche muscles/graisse aux déformations cinématiques obtenues par *skinning* ou tout autre technique équivalente. L'effet dynamique est calculé à travers une simplification des lois de l'élasticité linéaire. Pour cela, des éléments dynamiques associant les vibrations d'un ressort à un ensemble de poids assignés aux sommets du maillage permettent d'effectuer un *skinning dynamique*. L'emplacement, la fréquence et l'amplitude de ces effets sont spécifiés, de manière très simple, par l'utilisateur.

Sur cette couche vient se greffer une surface fine qui peut être soit la peau, soit directement un vêtement. Cette surface, d'aire quasiment constante, suit les déformations des muscles et des tissus graisseux tout en plissant lorsqu'elle se retrouve compressée. Pour cela, l'utilisateur positionne et oriente une courbe de contrôle qui se déforme à longueur constante et spécifie sa région d'influence. Pendant l'animation, des plis géométriques sont générés en temps-réel dans les zones couvertes par l'outil, le maillage étant raffiné localement à la volée.

Mots-clefs : modélisation, animation, temps-réel, plis, dynamique, inertie, personnage

Abstract

Traditional animation of characters relies on the design of an external shape, called the skin, and an internal skeleton, composed of a hierarchy of frames, which enables one to control the deformation of the skin in a consistent and predictable way. Since it is relatively efficient, this *skinning* technique is widely used in computer graphics production environments. However, it also has inherent limitations. Even if a large amount of research work were to solve the other known deficiencies of the technique, the fundamental issue remains that some types of deformations can never be generated by it. Examples of this are the dynamic wrinkles of skin or clothes near joints, or inertia effects of muscles and fatty tissues when sharp movements are involved. In this thesis, we propose two modeling and animation techniques which address those specific problems, and which can be used to enhance existing animation sequences.

Our first contribution addresses the addition of dynamic effects which are typical for the muscles/fatty tissues layer to kinematic deformations obtained through *skinning* or an equivalent technique. The dynamic effects are computed using a simplification of linear elasticity theory. To this end, dynamic elements which associate damped spring vibrations with a set of per vertex weights for the mesh, perform a so called *dynamic skinning*. The location, frequency and amplitude of the effects can be specified in a straightforward way by the user.

On top of this layer comes a thin surface which can either represent the skin or clothes. This surface of almost constant area follows muscle and fatty tissue deformations by wrinkling when it gets compressed. In order to guide this behavior the user interactively positions and orients a deformable control curve of constant length on a mesh, and specifies its region of influence. During the animation geometric folds are generated in real-time in the regions covered by the tool, and the mesh is locally refined on the fly where and when needed.

Keywords : modeling, animation, real-time, wrinkles, dynamics, inertia, character