Textures de dilatation pour la génération de plis

Jean Combaz **Fabrice Neyret**

iMAGIS, Laboratoire GRAVIR
INRIA Rhone-Alpes
38334 Saint Ismier Cedex

Jean.Combaz@imag.fr, Fabrice.Neyret@imag.fr
http://www-imagis.imag.fr/Membres/Jean.Combaz/

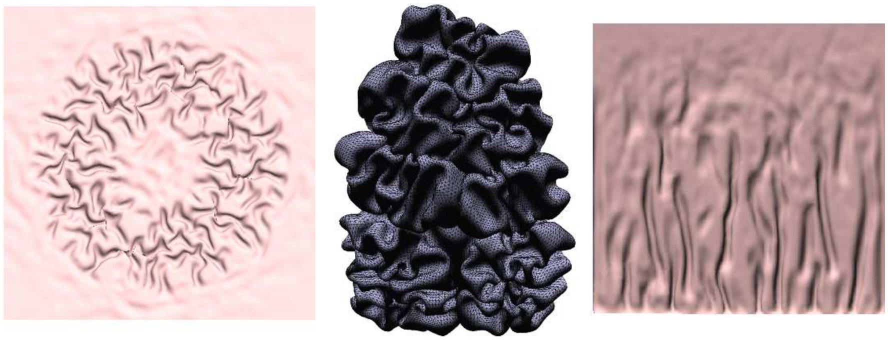

FIG. 1 – *Exemples de quelques plissements générés par notre outil.*

Résumé : *Nous introduisons ici les* **textures de dilatations** *pour ajouter des détails à une surface. Dans cet article nous nous intéressons plus particulièrement aux plis. L'utilisateur peint les attributs de dilatation sur la surface (l'intensité et la direction de dilatation, la longueur d'onde et la régularité des plis) à l'aide d'outils interactifs ou procéduraux. le système génère alors les plis, fronces ou cloques qui en résultent en calculant un nouvel équilibre de la surface. Les résultats montrent que cet outil permet au graphiste de contrôler facilement l'aspect des plis et drapés en ajoutant localement de la surface, ce qui est proche de la manière de penser des sculpteurs.*

Mots-clés : modélisation de surfaces, interface utilisateur, modélisation procédurale , plis, croissance, détails, imperfections.

1 Introduction

Les drapés et les plis peuvent se former dans de nombreuses situations (voir Figure 2), depuis l'action de la gravité ou des frottements sur les tissus jusqu'à la croissance de surfaces élastiques contraintes (par exemple de vieilles couches de peinture, le développement de surfaces biologiques ou géologiques). La séquence des actions qui a pu générer une surface plissée donnée peut être très complexe, voir inconnue (comme pour un lit défait) et l'état initial de la surface très artificiel (comme pour un vêtement). En conséquence, la simulation physique de ces objets, qui suppose la connaissance de l'état initial et des forces agissantes, est souvent difficile à mettre en oeuvre.

Pourtant les artistes traditionnels savent peindre et sculpter ces drapés, sans pour autant avoir à définir état initial et forces ni à simuler la physique. De plus différentes situations peuvent conduire à des formes similaires. ce qui autorise les artistes à se reposer sur de nombreuses intuitions pour interpréter les formes qu'ils ont dans l'esprit ou devant les yeux. Par exemple les artistes créent de nouvelles formes ou en modifient des existantes en ajoutant ou en supprimant de la matière plutôt qu'en considérant un état initial idéal et appliquant une série de forces pour la modéliser.

Notre but est alors de faciliter le travail du graphiste en lui permettant de 'peindre' des déformations comme les plis ou les drapés. Notre approche consiste à partir d'un état initial, qui est une approximation de la forme que l'on veut créer (comme un cylindre dans le cas de la nappe de la Figure 2, ou un corps pour des vêtements). Cela ressemble à la première ébauche du sculpteur ou du peintre. Puis nous introduisons un nouveau concept: ajouter de la matière pour la modélisation de surface en considérant des opérateurs de dilatation ou contraction, isotropes ou anisotropes. Le contrôle peut se faire soit de manière interactive, soit procéduralement. Notre système génère alors des plis en trouvant un nouvel équilibre pour la surface, ce qui est facilité par le fait que l'état initial est géométriquement proche de l'état final.

Notre système possède des similitudes avec un simulateur physique; il n'a cependant pas besoin de simuler la dynamique. De plus, nous raffinons la surface et effectuons les calculs seulement là ou c'est nécessaire, c'est à dire là où l'utilisateur ajoute des plis. Dans cet article nous montrons plusieurs applications de ce principe afin d'en illustrer l'utilité.

Notre article, qui est une reprise de notre article paru à Pacific Graphics cette année (cf. [CN02]), se compose de cinq sections. Le paragraphe 2 passe en revue les travaux existant sur la modélisation de détails et la simulation des plis. Puis nous décrirons notre concept de *texture de dilatation* du point de vue de l'utilisateur dans la section 3 ainsi que du point de vue technique dans la section 4. Nous discuterons alors nos résultats dans la section 5 puis nous conclurons.

2 État de l'art

Il existe de nombreuses manières de créer les détails géométriques nécessaires au réalisme des objets de synthèse. Cela inclut deux aspects: comment définir et contrôler ces détails, et comment les représenter.

La définition de ces détails peut se faire par:

- des outils interactifs qui permettent à l'utilisateur de définir explicitement les détails, comme les déformations de forme libre (FFD) [SP86, Coq90] ou de peindre directement ces détails sur la surface [HH90];
- des outils procéduraux qui permettent à l'utilisateur de contrôler les paramètres d'un générateur automatique de détails. Ce générateur peut être alors soit générique [Per85, EMP+94, FF80] soit spécialisé [FLCB95, PHM, Pru93, BB90, WNH97];
- des outils de simulation qui reproduisent des lois physiques. Ils sont particulièrement utilisés pour les vêtements [TF88, BHW94, BW98] ou la création de certaines structures biologiques [FMP92, WK91, Tur91].

Ces détails peuvent alors être représentés par:

- les moyens classiques pour représenter une surface comme les maillages polygonaux;
- les cartes de déplacement (littéralement displacement maps), i.e. textures qui encodent le relief et qui peuvent être converties en géométrie au moment du rendu [WMF+00, GSS99];
- les textures volumiques [KK89, Ney98] et les hypertextures [PH89] qui n'encodent pas les détails d'une manière surfacique;
- les cartes pour le plaquage de normales (littéralement bump maps) [Bli78] qui modifient l'aspect de la géométrie en ne changeant que l'illumination.

Des transitions entre ces différentes représentations sont définies dans [BM93, COM98].

L'idée de simuler une croissance a déjà été introduit dans le cadre d'objets biologiques [WFM01, PHM, Pru93]. La forme de surfaces élastiques comme les vêtements est généralement obtenue au moyen de simulations physiques [TF88, BHW94, BW98], mais certains outils géométriques peuvent être utilisés pour imiter la physique [DKT98]. Comme D'Arcy Thompson [Tho17] le suggère il y a de nombreuses approches possibles pour expliquer une forme donnée.

Les outils interactifs comme Maya Artisan sont pratiques pour l'utilisateur mais ils demandent la modélisation explicite des formes ce qui peut être très rébarbatif pour les détails. Les outils de simulation génèrent des formes réalistes, mais cela peut parfois prendre beaucoup de temps pour obtenir la forme finale. De plus l'utilisateur doit définir un état initial, les paramètres physiques du matériau ainsi que les forces agissantes, ce qui n'est pas toujours connu: un sculpteur reproduisant le modèle qu'il a dans son esprit ou devant les yeux ne peut pas le décrire facilement en temps qu'expérience de mécanique. Les outils procéduraux permettent un haut niveau de contrôle ce qui est très pratique pour l'utilisateur; par contre de tels outils ont surtout été proposés dans le cadre des textures [Per85, EMP+94] et seulement peu d'entre eux sont utilisés pour créer de la géométrie.

Notre but est de créer des formes géométriques ressemblant au résultat d'une simulation physique, tout en laissant à l'utilisateur un haut niveau de contrôle des détails générés, comme s'il utilisait un outil procédural. Nous allons surtout nous concentrer sur la génération de ces détails. Notre implémentation est pour le moment

FIG. 2 – *Quelques drapés et plis rencontrés dans la réalité: des vêtements réels ou sculptés, un chouchou, des plissements de lave, une nappe, de la peau, un vieille couche de peinture, des plis de macadam et une bâche en plastique posée sur le sol.*

limitée à une représentation sous forme de maillages polygonaux, mais on pourrait adapter notre modèle à d'autres représentations comme les cartes de déplacements ou de normales. Dans cet article, nous allons nous intéresser aux drapés et aux plis qui sont très utilisés dans dans l'art classique, mais aussi très difficiles à modéliser avec les outils actuels de la synthèse d'image.

3 Les *Textures de dilatation* du point de vue de l'utilisateur

Le principe est de permettre à l'utilisateur de contrôler les paramètres de haut niveau, soit globalement soit en peignant leurs variations sur la surface. Le solveur (qui est une partie de notre système de modélisation) modifie alors le maillage pour générer un nouvel équilibre de notre surface, en ajoutant des détails -des plis-. Ce résultat est calculé soit interactivement, soit en différé, selon la complexité de la tache. La Figure 3 illustre une session interactive.

Les premières poignées de contrôle correspondent à la dilatation: l'utilisateur contrôle la magnitude et la direction de la dilatation, et fournit aussi d'autres informations comme la longueur d'onde désirée ainsi que la régularité des plis. La dilatation est donnée par un champ de tenseurs, que l'on peut représenter localement par une ellipse (dilatation d'un cercle en ellipse). Dans le cas particulier d'une dilatation unidirectionnelle, on peut représenter ce champ de tenseurs par un champ de vecteurs.

Une autre série de poignées contrôle les degrés de liberté de la surface. Dans notre implémentation nous pouvons fixer la position de certaines zones de la surface (selon un axe, selon un plan ou totalement), rajouter des forces, prendre en compte la collision avec d'autres objets pour contraindre la surface.

Les paramètres qui correspondent aux premiers moyens de contrôle sont assez semblables aux shaders des outils de rendu puissants. Ils peuvent être spécifiés de différentes manières suivant qu'ils soient uniformes ou non sur la surface, et suivant la façon dont l'utilisateur préfère les contrôler. De plus ces paramètres peuvent être scalaires, vectoriels ou tensoriels. Dans notre implémentation nous laissons l'utilisateur les contrôler interactivement, procéduralement ou par l'intermédiaire de cartes (textures). Les cartes qui n'encodent pas des paramètres scalaires peuvent être considérés comme une collection de cartes scalaires (que l'on pourra éditer avec les logiciels classiques de dessin), ou comme des cartes vectorielles. Tout comme avec les shaders, le contexte d'utilisation

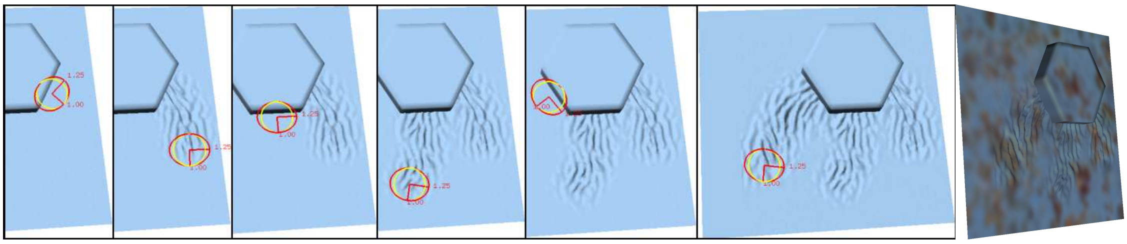

FIG. 3 – *dessin interactif de plis sur une couche de peinture autour d'un écrou (voir Figure 2). L'ellipse foncée représente la magnitude et la direction de la dilatation qu'applique l'outil à la surface. L'orientation de l'outil suit le chemin de la souris.*

est conditionné par la complexité et la qualité désirée. Une simple édition de la surface peut-être faite interactivement alors que les formes assez complexes ont intérêt à être construites grâce à des textures ou une définition procédurale.

4 Les *Textures de dilatation* du point de vue technique

Comme cela a été mentionné dans l'introduction, définir l'équilibre d'une surface connaissant les contraintes qui s'y exercent est un problème standard de mécanique (théorie des coques) pour lequel de nombreuses solutions existent pour sa résolution (par exemple grâce aux éléments finis). Nous décrivons ici notre implémentation d'un modèle simplifié qui utilise des techniques déjà existantes.

Premièrement nous utilisons un maillage triangulaire pour représenter la surface sur laquelle on veut rajouter les détails. Nous définissons un *état de référence* par une longueur l_0 pour chaque arête, et la courbure moyenne κ_0 pour chaque point. On considère les valeurs l_0 comme les longueurs au repos des arêtes, que l'on va modifier selon la croissance. Ceci est donc un état *virtuel* de référence, car on peut seulement l'atteindre localement (rien ne garantit l'existence réelle d'une telle surface). Après avoir appliqué la dilatation ou la contraction définie par l'utilisateur sur les l_0, un solveur itératif se charge de calculer un nouvel équilibre de la surface. Chaque itération déplace les points pour faire décroître les tensions dans la surface. Ces tensions -ou contraintes- sont obtenues en comparant localement la présente configuration de la surface avec l'état de référence. Par la suite nous noterons l la longueur d'une arête et κ la courbure moyenne mesurée sur la surface. Notre surface peut être orientée, autorisant une croissance selon une face privilégiée (par exemple si la surface est sensée être une frontière délimitant un volume). Nous définissons alors une normale orientée $\vec{N}$ en chaque point de la surface.

Voici un aperçu de notre algorithme qui sera décrit en détails dans les sections suivantes :

- application de la dilatation (ou de la contraction) par la modification des longueurs à vide l_0;
- optimisation locale du maillage virtuel (possibilité d'ajouter et de supprimer des arêtes);
- itérer des petits déplacements $\delta\vec{F}$ qui vont diminuer les contraintes.

Dans le cas de dilatations importantes, cet algorithme doit être adapté. Nous en reparlerons dans la section 4.5.

4.1 Dilatation

La dilatation agit sur la longueur au repos des arêtes: soit un tenseur de dilatation donné τ, alors l'arête $\vec{l_0}$ voit sa longueur multipliée par un facteur $\sqrt{\vec{l_0}^t \tau \vec{l_0}}$. En pratique on a un champ de tenseur de dilatation $\tau(u,v)$ et on intègre ce tenseur le long de l'arête pour avoir une valeur moyenne. On peut aussi utiliser une représentation MIP-mapping de la texture de dilatation.

4.2 Optimisation du maillage

Alors que le maillage initial peut être grossier, la génération de petits détails demande un maillage relativement fin. Nous subdivisons alors le maillage suivant la longueur d'onde des plis requise, ainsi que le gradient des paramètres locaux. De plus les triangles trop allongés sont éliminés afin de maintenir une triangulation de bonne qualité. Ceci est important en pratique pour des raisons de stabilité et de performance dans le solveur. Nous avons implémenté la permutation, la subdivision et l'effacement d'arêtes (voir Figure 4) en utilisant un critère inspiré des maillage de Delaunay appliqué aux surface gauches (cf [WW94]). On permute les arêtes pour rendre maximum le plus petit des angles des triangles. On subdivise les arêtes dont la longueur ou l'angle entre les normales de ses sommets excède un seuil (ces seuils sont définies par l'utilisateur), de plus on insère des points là où les gradients de dilatations sont trop importants afin de garder une bonne précision dans la simulation. L'effacement des arêtes est effectué dans le cas contraire et en utilisant des seuils plus petits afin d'assurer un hystérésis entre ces deux opérations (cela permet de garder une certaine stabilité dans la configuration du maillage).

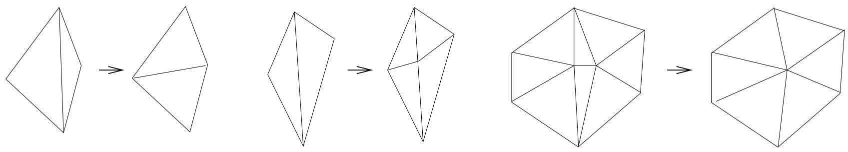

FIG. 4 – *Permutation, subdivision et effacement d'arêtes.*

4.3 Évaluation des contraintes

On calcule tout d'abord la déformation subie en passant de l'état de référence à l'état courant de la surface, l'état de référence étant définie par les longueurs à vide et l'état courant (déformé) par la position des points. On utilise pour cela le tenseur des déformations de Green-Lagrange (4.1) plutôt que celui de Cauchy car il est plus adapté aux grandes déformations mais reste assez simple à évaluer (cf. [DDCB01] pour la discussion des ces choix). Nous évaluons ce tenseur pour chaque triangle. Il est représenté par une matrice 2×2:

$$(\epsilon)_{ij} = \left(\frac{\partial x}{\partial \Omega_i} \cdot \frac{\partial x}{\partial \Omega_j} \right) - \delta_{ij} \tag{4.1}$$

où δ est la fonction de Kronecker [1] et (Ω_1,Ω_2) est un repère local de l'élément. Ce tenseur donne la déformation entre le triangle dans l'état de référence (défini par ses longueurs au repos) et l'état présent.

Le tenseur des contraintes σ peut alors en être déduit en utilisant la loi de Hook (en supposant que l'on a affaire à un matériau élastique linéaire et isotrope), permettant d'obtenir les forces en chaque sommet:

$$\sigma = \lambda tr(\epsilon) I + 2\mu\epsilon \tag{4.2}$$

où μ représente la rigidité de la surface et λ mesure son incompressibilité. Pour plus de détails, voir [OH99] où le problème a été exposé en 3D (éléments finis explicites). Puisque nous avons une surface gauche, nous nous sommes restreints à un problème à deux dimensions, puis l'avons adapté à des surfaces courbes. Les forces qui expriment la déformation tangentielle de la surface sont déduites de σ A chaque noeud i on somme les contributions $\vec{F_{ki}}$ des triangle k qui ont le sommet i en commun, où:

$$\vec{F_{ki}} = -\frac{a_{0_k}}{2} \sum_{j=1}^{3} \vec{x_j} \sum_{\alpha=1}^{2} \sum_{\beta=1}^{2} (L^{ki})_\alpha (L^{kj})_\beta (\sigma)_{\alpha\beta} \tag{4.3}$$

a_{0_k} est l'aire du triangle k, L^{kj} la représentation sous forme de vecteurs de sa fonction de base linéaire associée au sommet j (ce vecteur dépend uniquement des longueurs au repos) et $\vec{x_j}$ la position 3D du sommet j. La somme des $\vec{F_{ki}}$ est alors projetée dans le plan tangent.

Puis nous ajoutons des forces de courbure $\vec{F_\kappa}$ et des forces normales de contrainte $\vec{F_N}$. $\vec{F_\kappa}$ est une force de rappel qui tend à lisser la surface en limitant la différence de courbure avec l'état de référence. $\vec{F_N}$ est une force créatrice de plis qui traduit la compression de la surface en un déplacement suivant la normale. C'est aussi grâce à cette force que l'on contrôle la forme des plis. Afin d'éviter la complexité de la théorie des coques, nous avons utilisé des simplifications inspirées par [DMSB99]. Nous avons choisi :

$$\vec{F_\kappa} = -k_\kappa.(\kappa - \kappa_0)\vec{N} \tag{4.4}$$

$$\vec{F_N} = (k_p \mathrm{f}(\kappa - \kappa_0) + k_{p_i}) C_a \vec{N} \tag{4.5}$$

1. δ_{ij} est égale à 1 si i=j et à 0 sinon.

κ et $\vec{N}$ sont calculés en utilisant la même interpolation des points voisins que [DMSB99]. k_κ and k_p sont deux constantes. k_{p_i} est un biais qui permet de pousser la surface dans une direction privilégiée ($\vec{N}$ or -$\vec{N}$). Si aucune direction n'est préférée, k_{p_i} peut être égale à zéro, mais il est préférable de la fixer à de faibles valeurs aléatoires pour chaque points: cela évite des problèmes dans le cas de surfaces rigoureusement plates. C_a est le taux de compression surfacique de la cellule entourant le point considéré. On a:

$$C_a = \frac{\sum_{cellule} a_{0_i} - a_i}{\sum_{cellule} a_{0_i}} \tag{4.6}$$

où a_0 et a sont les aires des triangles dans l'état au repos et courant (déduites des longueurs des arêtes l_{0_i} et l_i). $f(\kappa)$ est une fonction qui contrôle la forme des plis; nous expliquons comment la choisir dans la section suivante.

Notons que toutes ces forces sont relativement indépendantes de la discrétisation: elles sont évaluées plus précisément dans le cas de triangles plus petits et presque équilatéraux, mais il n'y a pas de biais. Si une surface en équilibre est subdivisée, sa forme va être conservée (si l'on excepte quelques petites erreurs numériques).

4.4 Contrôle de la forme des plis

On introduit $\kappa^*(0) = \kappa - \kappa_0$ et $\frac{\partial \kappa^*}{\partial t} = L(\kappa^*)$ où L est le noyau d'un filtre de diffusion anisotrope non uniforme (cf [DF95] pour plus d'information sur les filtres différentiels). On choisit alors $f(\kappa - \kappa_0) = \sigma(\kappa^*)$ où $\sigma()$ est une sigmoïde[2]. Pour des κ^* assez grands nous voulons juste savoir dans quelle direction la surface doit être poussée. Maintenant illustrons le choix du filtre en 1D en prenant

$$L(u) = \frac{\partial u^2}{\partial x^2} g(u + \nu^2 \frac{\partial u^2}{\partial x^2}) \tag{4.7}$$

où $g(x) = \frac{x^2}{1+\alpha x^2}$. La fonction g permet de lisser plus ou moins un signal u suivant sa fréquence (c'est un filtre passe bande: le signal n'est pas lissé s'il a pour pulsation ν). A l'équilibre κ^* reste stationnaire, donc $L(\kappa^*) = 0$. Cela arrive si $\kappa^* + \nu^2 \frac{\partial^2 \kappa^*}{\partial x^2} = 0$, i.e. pour un signal harmonique de pulsation ν. Ainsi si nous voulons des plis d'une longueur d'onde λ, on doit choisir $\nu = \frac{\lambda}{2\pi}$. Remarquons que si κ^* est nul, il est aussi stationnaire, d'où la nécessité de prendre k_{p_i} non nul pour initier un premier mouvement et créer de la courbure (et rendre ainsi κ^* différent de zéro). En fait nous utilisons un opérateur anisotrope, c'est à dire que nous appliquons le même filtre (4.7) mais dans deux directions et avec deux valeurs de ν différentes. Dans une première direction (celle du plis) $\nu_1(\Lambda)$ va contrôler sa régularité, et dans la direction orthogonale $\nu_2(\lambda)$ contrôlera la longueur d'onde des plis (cf Figure 5).

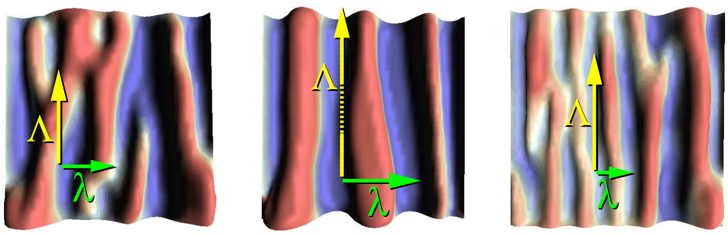

FIG. 5 – *Réglage de la longueur d'onde λ et de la régularité Λ.*

4.5 Modification de l'algorithme dans le cas de grandes dilatations

Si le taux de dilatation est raisonnable, augmenter la longueur au repos des arêtes en un seul pas est suffisant. Sinon il vaut mieux itérer, c'est à dire appliquer successivement des dilatations partielles. Et dans le cas de très grandes dilatations (pour la morphogénèse, ce qui dépasse le sujet de cet article) il est nécessaire d'ajouter de la plasticité afin d'éviter que des contraintes qui n'arrivent pas à se dissiper ne s'accumulent. La plasticité revient à atténuer la mémoire que l'on avait de l'état initial. Cela est réalisé en relaxant la longueur au repos: à chaque fois que l'on obtient un équilibre, l_0 est remplacée par $(1 - \varepsilon)l_0 + \varepsilon l$. Cela peut aussi être utile d'ajouter de la plasticité dans le cadre de la modélisation de détails, surtout si la texture de dilatation est trop complexe pour que les contraintes ne se dissipent, sinon les plis peuvent sembler trop distordus.

2. Une sigmoïde est une fonction monotone croissante variant entre -1 et 1 avec une transition rapide en 0.

4.6 Optimisations

Dans notre implémentation nous utilisons plusieurs classes d'optimisations:

- Comme nous l'avons mentionné dans la section 4.2, nous adaptons le maillage au besoin de la simulation.
- Notre solveur utilise différents pas de déplacements pour chaque points, ceci afin de consacrer du temps de calcul seulement là où c'est nécessaire. Cette adaptation a été faite dans l'esprit de [DDCB01]: on calcule le pas maximum tolérable pour chaque point. Soit δ le plus petit d'entre eux. Si un point nécessite un pas $\delta_i \in [2^n\delta, 2^{n+1}\delta[$, nous ne recalculons ses forces que pour les itérations dont le numéro est un multiple de 2^n.
- Dans le cas d'une utilisation interactive, nous définissons un morceau de surface actif en dehors duquel aucun calcul n'est effectué: l'outil interactif dilate ou contracte une surface circulaire limitée; Nous considérons alors une portion de surface circulaire légèrement plus grande dans laquelle l'aspect de la surface pourrait être affecté. On active alors la simulation que dans cette zone. Un décompteur est attaché à chaque point de la zone et l'on désactive les points actifs dont le compteur a atteint zéro.

5 Résultats

Dans la Figure 6, nous montrons une dilatation unidirectionnelle uniforme d'une surface carrée dont les bords gauche et droite sont attachés. Nous avons fait l'expérience avec une petite longueur d'onde et une petite régularité, et avec une plus grande longueur d'onde et régularité. En comparaison, on peut voir sur la gauche une bâche en plastique réelle. Sur la droite, des fronces ont été simplement obtenues en peignant d'étroites bandes dans la texture de dilatation. Dans l'image de droite de la Figure 1 nous avons utilisé un taux croissant de dilatation du haut vers le bas et introduit une force de gravité ainsi qu'une détection de collision avec le sol (un plan horizontal), tous les bords étant attachés: cela produit des plis positifs, séparés de zones relativement plates.

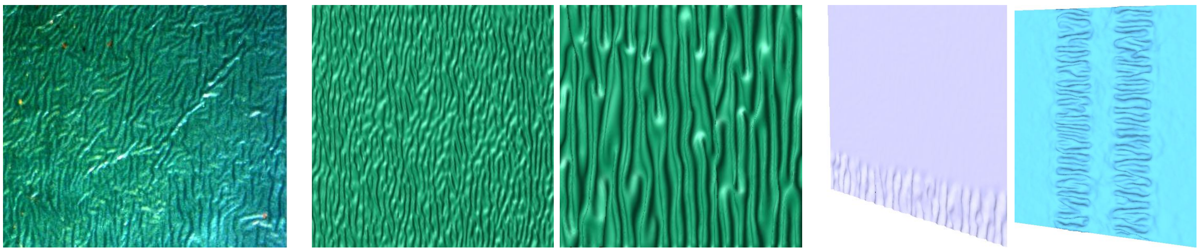

FIG. 6 – *Plis réguliers (à gauche une image d'une bâche en plastique réelle); Fronces.*

Dans la Figure 7, une dilatation circulaire unidirectionnelle est appliquée à une couronne de surface (la texture de dilatation est montrée à gauche), avec plusieurs longueurs d'ondes et différentes régularités: sur la gauche la longueur d'onde et la régularité sont pratiquement nulles, ce qui donne un motif presque aléatoire. Sur la droite elles sont plus importantes et le motif montre alors des plis beaucoup plus organisés.

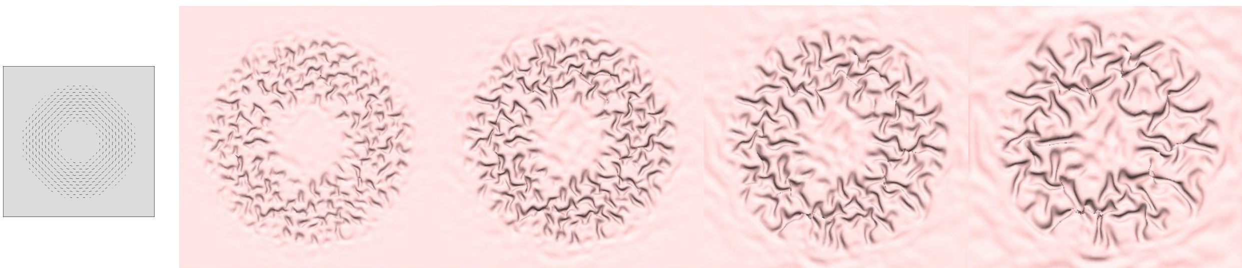

FIG. 7 – *Plis sur une couronne de surface*

La Figure 8 montre un chouchou calculé avec différents taux de dilatation. Il est obtenu en dilatant un tore comme indiqué dans la Figure de gauche. Comme dans la réalité, la surface doit être contrainte pour éviter d'obtenir un simple tore plus grand: dans la réalité un anneau en caoutchouc est insèré à l'intérieur; Nous réalisons l'équivalent en rajoutant un tore rigide à l'intérieur de notre surface torique mobile (i.e. une détection de collision empêche le grand tore de trop s'agrandir au lieu de plisser). D'autres solutions peuvent être utilisées dans le même but: dans l'image du milieu de la Figure 1 nous avons défini plusieurs bandes rigides sur la surface (ce qui est plus ou moins équivalent avec ce qui se fait avec les vrais vêtements). On peut remarquer que malgré le fait qu'il n'y ait pas de tests d'auto-collision, le lissage de la courbure suffit à éviter les collisions locales (une déformation purement géométrique comme avec les textures de déplacement n'aurait pas pu le faire). Mais si l'on ne fait aucun test, des plis distants peuvent néanmoins s'intersecter. Cependant cela n'apparaît que quand le taux de dilatation dépasse un facteur de 3 (cf Figure 9). Une dilatation plus importante du chouchou dans la Figure 8 aurait probablement montré des auto-collisions.

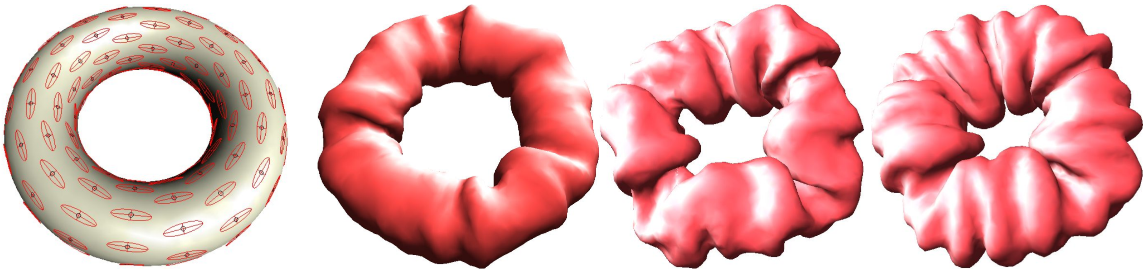

FIG. 8 – *Un chouchou obtenu à partir d'un tore*

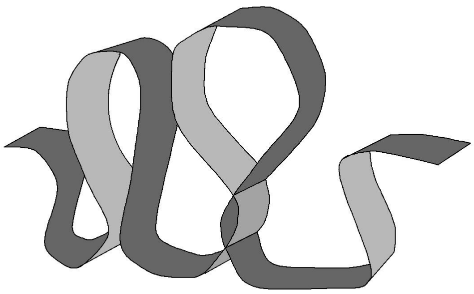

FIG. 9 – *Pour des dilatations importantes (typiquement supérieur à 3) des plis voisins peuvent s'intersecter.*

Les temps de calcul avec notre implémentation vont de quelques secondes dans le cas d'une édition interactive (cf Figure 3) à plus d'une demi-heure sur un Pentium III à 700 MHz pour l'image du milieu de la Figure 1 (le maillage compte à peu près 50000 points). De meilleurs performances devraient être obtenues avec un solveur plus efficace, par exemple basé sur des méthodes implicites (voir [BW98, DMSB99]).

Bien que les objectifs de cet article soient limités à la modélisation de détails sur une surface, nous avons expérimenté des textures de dilatation plus fantaisistes, qui affectent la forme de la surface dans sa globalité. Dans nos travaux futurs, cela nous permettra d'aller vers des méthodes de morphogénèse pour modéliser des formes complexes. Nous présentons ici quelques résultats préliminaires: la Figure 10 montre une forme ressemblant à un cerveau obtenue à partir d'une dilatation isotrope sur une sphère. Une texture (montrée à droite) contenant des taches de dilatation isotropes est appliquée à un carré, et donne l'ensemble des cloques au milieu de la Figure 10. Un autre exemple de résultat montre dans la Figure 11 un carré se courber après avoir été soumis à une dilatation isotrope limitée à certaines zones (La texture correspondante est visible au milieu).

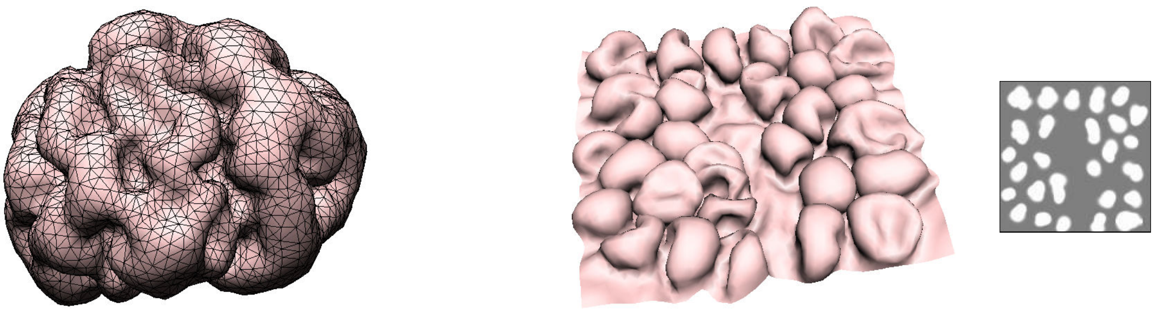

FIG. 10 – *Circonvolutions et cloques*

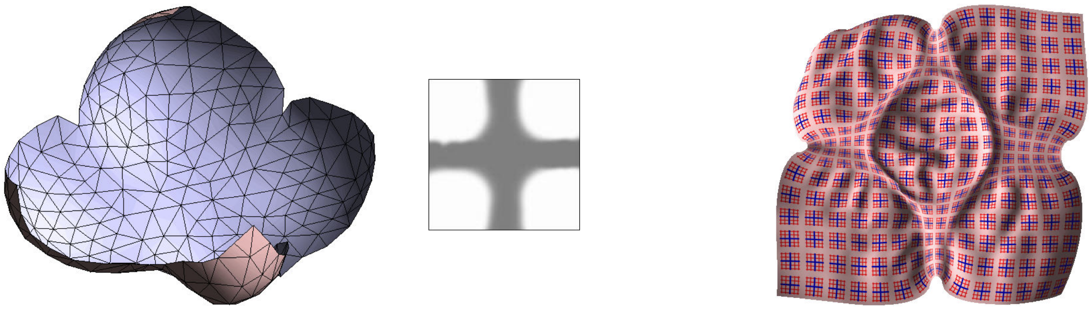

FIG. 11 – *Croissance non uniforme d'un carré*

6 Conclusion

Nous avons introduit le concept des *textures de dilatation*, lesquelles permettent à l'utilisateur de spécifier l'aspect des plis sur une surface avec un contrôle à haut niveau (i.e. sans avoir à définir explicitement les plis comme cela serait fait avec des cartes de déplacement ou une modélisation directe). L'utilisateur spécifie les zones de plis, un taux et une direction de dilatation et les possibles contraintes qui peuvent s'exercer sur la surface. L'utilisateur peut utiliser un outil interactif de dessin des plis, ou passer par une carte des dilatations (qu'il définit soit explicitement, soit procéduralement). Bien que le calcul de l'équilibre possède beaucoup de similitudes avec les solveurs de simulation physique, notre approche est plus compatible avec les connaissances et les souhaits que peut avoir un artiste: l'utilisateur n'a besoin de connaître que le genre de forme qu'il veut obtenir, et non pas l'historique des forces qu'il faudrait appliquer en réalité pour obtenir le résultat (beaucoup de sculptures baroques ont d'ailleurs des plis très exagérés qui ne sont probablement pas physiques, et même des surfaces non développables).

En ce qui concerne les travaux futurs, nous allons inclure les auto-collisions dans notre implémentation pour empêcher les plis de s'intersecter. Nous voudrions aussi étudier la génération direct de textures pour le placage de normales (afin d'éviter d'obtenir une géométrie trop lourde, si l'objet doit être observé de loin). Notre but à plus long terme est d'expérimenter la morphogénèse, c'est à dire le modélisation de formes qui sont principalement le résultat de phénomènes de croissance. Comme cela implique des taux de dilatation d'un autre ordre, nous sommes particulièrement intéressé par la définition procédurale de textures de dilatation, qui auraient par exemple des propriétés fractales.

Références

[BB90] Norman I. Badler and Welton Becket. Imperfection for realistic image synthesis. *Journal of Visualization and Computer Animation*, 1(1):26–32, August 1990.

[BHW94] David E. Breen, Donald H. House, and Michael J. Wozny. Predicting the drape of woven cloth using interacting particles. In *Proceedings of SIGGRAPH 94*, pages 365–372. ACM SIGGRAPH / ACM Press, July 1994.

[Bli78] James F. Blinn. Simulation of wrinkled surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, volume 12(3), pages 286–292, August 1978.

[BM93] Barry G. Becker and Nelson L. Max. Smooth transitions between bump rendering algorithms. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 183–190, August 1993.

[BW98] David Baraff and Andrew P. Witkin. Large steps in cloth simulation. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 43–54. ACM SIGGRAPH / Addison Wesley, July 1998.

[CN02] Jean Combaz and Fabrice Neyret. Painting folds using expansion textures. In *Pacific Graphics 2002 Proceedings*, pages 176–183, October 2002.

[COM98] Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-preserving simplification. *Proceedings of SIGGRAPH 98*, pages 115–122, July 1998.

[Coq90] Sabine Coquillart. Extended free-form deformation: A sculpturing tool for 3D geometric modeling. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 187–196, August 1990.

[DDCB01] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of ACM SIGGRAPH 2001*, pages 31–36. ACM Press / ACM SIGGRAPH, August 2001.

[DF95] Rachid Deriche and Olivier Faugeras. Les EDP en traitement des images et vision par ordinateur. Technical Report RR-2697, Inria, Institut National de Recherche en Informatique et en Automatique, 1995.

[DKT98] Tony D. DeRose, Michael Kass, and Tien Truong. Subdivision surfaces in character animation. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, pages 85–94. ACM SIGGRAPH, July 1998.

[DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of SIGGRAPH 99*, pages 317–324. ACM SIGGRAPH, August 1999.

[EMP+94] David Ebert, Kent Musgrave, Darwyn Peachey, Ken Perlin, and Worley. *Texturing and Modeling: A Procedural Approach*. Academic Press, October 1994. ISBN 0-12-228760-6.

[FF80] Alain Fournier and Don Fussell. Stochastic modeling in computer graphics. *Computer Graphics (Proceedings of SIGGRAPH 80)*, 14(3):108, July 1980. Held in Seattle, Washington.

[FLCB95] Kurt W. Fleischer, David H. Laidlaw, Bena L. Currin, and Alan H. Barr. Cellular texture generation. *Computer Graphics*, 29(Annual Conference Series):239–248, 1995.

[FMP92] Deborah R. Fowler, Hans Meinhardt, and Przemyslaw Prusinkiewicz. Modeling seashells. In Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 379–388, July 1992.

[GSS99] Igor Guskov, Wim Sweldens, and Peter Schröder. Multiresolution signal processing for meshes. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, pages 325–334, Los Angeles, California, August 1999. ACM SIGGRAPH / Addison Wesley Longman.

[HH90] Pat Hanrahan and Paul E. Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH 90 Proceedings)*, volume 24, pages 215–223, August 1990.

[KK89] James T. Kajiya and Timothy L. Kay. Rendering fur with three dimensional textures. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23(3), pages 271–280, July 1989.

[Ney98] Fabrice Neyret. Modeling animating and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):55–70, January–March 1998. ISSN 1077-2626.

[OH99] J.F. O'Brien and J.K. Hodgins. Graphical modeling and animation of brittle fracture. In *SIGGRAPH'99 Conference Proceedings*, pages 137–146. ACM SIGGRAPH, 1999.

[Per85] Ken Perlin. An image synthesizer. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19(3), pages 287–296, July 1985.

[PH89] Ken Perlin and Eric M. Hoffert. Hypertexture. In Jeffrey Lane, editor, *Computer Graphics (SIGGRAPH '89 Proceedings)*, volume 23(3), pages 253–262, July 1989.

[PHM] Przemyslaw Prusinkiewicz, Mark Hammel, and Radomír Mech. Visual models of morphogenesis: A guided tour. http://www.cpsc.ucalgary.ca/Redirect/bmv/vmm-deluxe/.

[Pru93] Przemyslaw Prusinkiewicz. Modelling and visualization of biological structures. *Graphics Interface '93*, pages 128–137, May 1993. Held in Toronto, Ontario, Canada.

[SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 151–160, August 1986.

[TF88] D. Terzopoulos and K. Fleisher. Modeling inelastic deformation: Viscoelasticity, plasticity, fracture. In *SIGGRAPH'88 Conference Proceedings*, pages 269–278, 1988.

[Tho17] D'Arcy Wentworth Thompson. *On Growth and Form*. Cambridge University Press, Cambridge, 1917.

[Tur91] Greg Turk. Generating textures for arbitrary surfaces using reaction-diffusion. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 289–298, July 1991.

[WFM01] Marcelo Walter, Alain Fournier, and Daniel Menevaux. Integrating shape and pattern in mammalian models. In *Proceedings of SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 317–326. ACM Press / ACM SIGGRAPH, August 2001.

[WK91] Andrew Witkin and Michael Kass. Reaction-diffusion textures. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 299–308, July 1991.

[WMF+00] Xiaochuan Corina Wang, Jérome Maillot, Eugene L. Fiume, Victor Ng-Thow-Hing, Andrew Woo, and Sanjay Bakshi. Feature-based displacement mapping. *Rendering Techniques 2000: 11th Eurographics Workshop on Rendering*, pages 257–268, June 2000.

[WNH97] Tien-Tsin Wong, Wai-Yin Ng, and Pheng-Ann Heng. A geometry dependent texture generation framework for simulating surface imperfections. In *Eurographics Rendering Workshop 1997*, pages 139–150, St. Etienne, France, June 1997. Eurographics / Springer Wien. ISBN 3-211-83001-4.

[WW94] William Welch and Andrew Witkin. Free-form shape design using triangulated surfaces. In *Proceedings of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, pages 247–256. ACM SIGGRAPH / ACM Press, July 1994.