



HAL
open science

Peau Implicite Active pour l'Habillage de Modèles Très Déformables

Mathieu Desbrun, Marie-Paule Cani

► **To cite this version:**

Mathieu Desbrun, Marie-Paule Cani. Peau Implicite Active pour l'Habillage de Modèles Très Déformables. 5ème Journées de l'Association Française d'Informatique Graphique (AFIG'97), Dec 1997, Rennes, France. inria-00537526

HAL Id: inria-00537526

<https://inria.hal.science/inria-00537526v1>

Submitted on 18 Nov 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.

Peau Implicite Active pour l’Habillage de Modèles Très Déformables

Mathieu Desbrun Marie-Paule Cani-Gascuel

iMAGIS[†]-GRAVIR / IMAG

BP 53, F-38041 Grenoble cedex 09, France

Mathieu.Desbrun@imag.fr

Marie-Paule.Gascuel@imag.fr

Résumé : *Cet article propose un nouveau modèle de surface déformable pour l’animation, appelé peau implicite active. Conçue pour habiller des modèles très déformables de type liquides visqueux modélisés par des systèmes de particules, cette peau suit automatiquement un objet cible, à l’instar des snakes utilisés en vision. Une formulation implicite reposant sur un potentiel discret permet de gérer efficacement les changements de topologie (séparations, fusions) de la surface, pour laquelle un dispositif de discrétisation très rapide est fourni. Outre son application à l’habillage de systèmes de particules, on peut utiliser ce modèle de peau déformable pour calculer des métamorphoses automatiques entre formes quelconques.*

Mots-clés : Surfaces déformables; Surfaces implicites; Potentiel discret; Tension de surface

1 Introduction

Les modèles d’objets très déformables développés en graphique reposent pour la plupart sur l’utilisation de systèmes de particules en interaction [TPF89, LJR⁺91, Ton91, LHVD95]. Différentes techniques peuvent être utilisées pour remplacer l’affichage d’un ensemble de sphères par un rendu visuel de qualité [Hab97]. Dans le cadre de l’animation de substances allant de la pâte à modeler aux liquides visqueux, nos modèles précédents [DG95, CGD97] utilisaient une surface implicite engendrée par les particules : chacune de ces dernières portait un potentiel décroissant avec la distance, la surface étant définie comme une isopotentielle pour la somme de ces contributions. Cependant, ce modèle n’était pas entièrement satisfaisant pour plusieurs raisons. Tout d’abord, le coût de calcul de la surface implicite, qui devait être échantillonnée à chaque pas de l’animation, devenait rhédibitoire dès que le nombre de particules dépassait plusieurs dizaines. En second lieu, ce modèle de surface était purement passif, sans aucune “inertie”, même si des déformations locales pouvaient lui être appliquées lors de collisions. Ainsi, la forme d’une goutte d’eau – modélisées par une seule particule – en chute libre était tout simplement sphérique. Enfin, cette couche implicite cachait mal le granularité du modèle interne, n’étant munie d’aucune tension de surface permettant de lisser la forme obtenue. Ce problème devient crucial lorsque la simulation des particules se fait de manière adaptative, le nombre de particules (et donc le volume de matière qu’elles représentent) pouvant changer au cours du temps [Des97]. De brusques changements de forme de la surface implicite se produiront alors à chaque modification de la structure interne.

Ne serait-ce que pour leur gestion automatique des changements de topologie comme les séparations et les fusions, les surfaces implicites restent cependant le modèle le mieux adapté à nos besoins. Cet article propose un nouveau modèle de peau alliant surfaces implicites et comportement dynamique avec inertie. Cette peau est définie comme l’isosurface d’un potentiel discret, dont les valeurs sont stockées sur les sommets d’un maillage de l’espace. Le potentiel est animé au cours du temps de manière à ce que la peau vienne épouser un modèle physique interne tout en simulant une tension de surface. Le modèle offre une polygonalisation très rapide et gère parfaitement les changements de topologie.

Après avoir brièvement passé en revue les techniques précédentes pour la définition d’une “peau” habillant un modèle interne en Section 2, nous présentons dans la Section 3 notre modèle de peau implicite mouvante. La section 4 détaille les détails pratiques à régler pour sa mise en œuvre. Des extensions permettant l’animation de matériaux se déformant à surface extérieure ou à volume constant sont proposées dans la section 5. Enfin, la Section 6 présente nos résultats pour l’habillage de systèmes de particules ainsi qu’une application à la métamorphose entre formes de topologie quelconque.

[†]. iMAGIS est un projet commun entre le CNRS, l’INRIA, l’Institut National Polytechnique de Grenoble et l’Université Joseph Fourier.

2 Approches antérieures de surfaces déformables

Le problème de l’habillage du modèle physique d’un objet par une *surface déformable*, à partir de laquelle seront effectués les calculs de rendu, s’est posé de longue date en animation de synthèse. Nous nous intéressons ici aux approches qui permettent d’associer des propriétés physiques à cette peau, que nous regroupons en deux grandes classes.

2.1 Approches Lagrangiennes

Une première approche, probablement la plus intuitive, consiste à discrétiser la peau en un ensemble d’éléments finis. Différentes méthodes à base de physique, telles que les réseaux masse-ressort ou les systèmes de particules par exemple, peuvent être utilisées pour calculer l’évolution de la surface au cours du temps [TPBF87, LJR⁺91, GVP91, Tur95].

Pour l’application qui nous intéresse, une telle approche pose problème : en effet, animer une surface de topologie fixe n’est pas envisageable lorsqu’il s’agit d’habiller des matériaux non structurés, pouvant changer de topologie. La conservation d’une polygonisation cohérente au cours du temps pose problème [LB94]. Une solution serait de représenter la peau par un nuage de particules restant à la surface d’un objet [LJR⁺91, WH94], mais puisque nous souhaitons disposer d’une surface à afficher, cela nécessiterait le calcul coûteux d’une nouvelle triangulation à chaque pas de l’animation.

2.2 Approches Euleriennes

Bien que ce soit relativement inhabituel, le concept “approche Eulerienne” peut être utilisé pour l’animation de surfaces déformables. Au lieu de discrétiser la matière en éléments, puis de suivre leurs mouvements, il s’agit de partitionner l’espace en volumes fixes, puis de regarder ce qui entre et sort de chacun de ces volumes.

Un modèle d’habillage particulier utilise ce type d’approche, mais avec une formulation originale : l’écran d’épingles d’Arash Habibi [Hab97]. Cette technique permet bien plus que de représenter une simple surface : elle offre la possibilité d’un véritable rendu volumique, même si sa mise en œuvre actuelle n’est que 2D.

D’une manière générale, le formalisme Eulerien semble bien adapté à un traitement rapide d’une part de l’évolution d’une surface, et d’autre part de sa représentation, grâce à des *voisinages fixes* connus. La phase la plus coûteuse des approches Lagrangiennes, à savoir le calcul à chaque pas de temps des nouvelles relations de voisinages entre points d’échantillonnage, est ainsi évitée.

2.3 Discussion

Dans l’optique d’un rendu surfacique performant et rapide, il semble que les approches Euleriennes soient bien mieux adaptées à notre application, bien qu’aucun modèle véritablement surfacique n’ait jamais été mis au point avec ce type d’approche. Leur avantage majeur est qu’elles conservent intrinsèquement la notion de *voisinage* entre points d’échantillonnage de par leur découpage fixe de l’espace, contrairement aux approches Lagrangiennes qui obligent une recherche de voisins coûteuse en temps de calcul. Cette particularité promet *a priori* de permettre une polygonalisation rapide même lors de changements de topologie. En conséquence, nous avons choisi une approche Eulerienne pour définir le modèle de peau qui fait l’objet de cet article.

3 Peau Implicite Mouvante

Dans cette section, nous nous attachons à décrire comment modéliser et animer de manière efficace une peau implicite ayant les qualités requises. Les détails de mise en œuvre seront vus dans la section suivante.

3.1 Définition d’une surface implicite mouvante

La définition formelle d’une surface implicite S qui bouge, se déforme, se coupe au cours du temps peut s’écrire fort simplement :

$$S(t) = \{ \mathbf{X} \in \mathbb{R}^3 / f(\mathbf{X}, t) = 0 \}.$$

où f est une fonction dite “potentiel” de $\mathbb{R}^3 \times \mathbb{R}$ dans \mathbb{R} . Nous avons fait l’hypothèse d’une isovaleur nulle pour simplifier les explications qui suivent. Dans le reste de cet article, nous prendrons la convention de définir l’intérieur d’un objet par le lieu où f prend des valeurs positives, et de diriger les normales unitaires \mathbf{n} vers l’extérieur. On a alors la relation suivante : $\mathbf{n} = -\nabla f / \|\nabla f\|$.

3.2 Intérêt d’un champ discret

Dans la section 2.2, on a vu que l’intérêt principal des approches Euleriennes était de faciliter la polygonalisation grâce à des relations de voisinage fixes. On peut dans cet esprit songer à utiliser une *fonction implicite discrète*: le potentiel n’a alors aucune formulation analytique, mais est simplement stocké sur une grille régulière. Une version continue peut facilement être interpolée par une fonction tri-linéaire à l’intérieur de chaque cube de la grille par exemple.

Cette formulation apporte bon nombre d’avantages. En premier lieu, le coût de l’évaluation du potentiel d’un point de l’espace se fait en temps constant, *indépendamment de la complexité de la surface manipulée*. Cette qualité permet de garantir un temps de traitement borné quelque soit la surface simulée.

En second lieu, la discrétisation de la surface engendrée par un potentiel discret est largement facilitée par rapport au cas général des surfaces implicites quelconques. Puisque l’on dispose déjà d’une grille régulière où le potentiel est stocké, les cubes par lesquels passe la surface sont facilement trouvés en recherchant les endroits où le potentiel change de signe. Cela correspond en tout point à l’algorithme des “Marching Cubes” [WMW86, LC87, FV97] sauf que les parties les plus longues, à savoir calcul du potentiel et traque de la surface, sont ici inutiles. Une fois ces cubes trouvés, on obtient une discrétisation en interpolant entre les sommets du cube afin de trouver des points de potentiel nul. Notre choix d’interpolation tri-linéaire permet d’effectuer la recherche de points à la surface de façon extrêmement simple : la position du point de potentiel nul entre deux sommets de signes opposés se fait par une simple règle de trois, *sans aucune dichotomie*.

Enfin, un dernier avantage apporté par cette formulation discrète est la possibilité d’un *contrôle localisé* de la surface. On peut en effet agir sur chaque valeur du potentiel discret pour affecter très localement la surface avoisinante, sans pour autant augmenter la complexité de la formulation.

3.3 Comportement différentiel du champ pour une surface animée

Puisque le potentiel définit une surface mouvante, il suit une équation différentielle. En écrivant que le chemin d’un point $\mathbf{X}(t)$ restant sur la surface au cours du temps a la propriété de vérifier $f(\mathbf{X}(t), t) = 0$ pour tout t , on obtient :

$$\forall \mathbf{X}(t) \in S(t), \quad \frac{df}{dt}(\mathbf{X}(t), t) = \frac{\partial f}{\partial t}(\mathbf{X}(t), t) + \nabla f(\mathbf{X}(t), t) \cdot \frac{d\mathbf{X}(t)}{dt} = 0 \quad (1)$$

Une façon d’animer la surface est alors de définir un champ de vitesses (puisque c’est la seule inconnue dans l’équation différentielle) et d’intégrer l’équation (1) au cours du temps. Ce type d’approche a d’ailleurs déjà été utilisé par Witkin et Heckbert [WH94] pour une application de type échantillonnage interactif, ainsi qu’en animation de textures [SS96] ou en vision [MSV95, Whi95].

Contrôler l’évolution de la surface revient donc à affecter à chaque point de la surface une *vitesse instantanée* de déplacement. Par intégration de l’équation différentielle (1), *la valeur du potentiel f sera modifiée pour prendre en compte le déplacement souhaité*. Ce contrôle *indirect* du mouvement de la surface, par le biais du potentiel la définissant, peut paraître une méthode quelque peu tortueuse. On verra pourtant par la suite que cette approche offre bien des avantages.

3.4 Définition d'une stratégie pour la peau

Pour donner finalement vie à la peau, il faut et il suffit de définir un *champ de vitesses* à chaque instant qui provoquera le mouvement souhaité. Cette section s'attache donc à proposer une *stratégie* pour la peau.

3.4.1 Attraction vers une surface cible

Notre peau est sensée recouvrir un modèle physique intérieur, comme un système de particules par exemple. Utiliser un enrobage implicite purement géométrique [CGD97], qui définit naturellement une *frontière* entre l'intérieur et l'extérieur, n'est pas suffisant dans le cas général puisqu'il ne peut simuler une tension de surface, et souffre de plus de discontinuités temporelles, visuellement désagréables, dans le cas de systèmes adaptatifs ou à niveaux de détail, où le nombre d'éléments physiques peut varier au cours du temps [Des97].

Une idée immédiate est d'alors utiliser cette iso-densité comme *cible*, c'est-à-dire que la peau sera constamment *attirée vers cette frontière*. On contourne ainsi le problème de l'habillage purement géométrique en introduisant une simulation de surface déformable tendant à englober un volume donné. De plus, la stratégie permettant ce comportement peut s'écrire très simplement :

$$\frac{d\mathbf{X}}{dt}(t)_{cible} = \alpha (G_{cible}(\mathbf{X}) - iso) \mathbf{n}(\mathbf{X}) \quad (2)$$

où α est un scalaire, G_{cible} la densité de masse induite par le modèle interne, et iso l'iso-densité choisie. La signification de cette équation est assez intuitive : si la surface actuelle est à l'intérieur de la matière, elle devra gonfler selon sa normale pour rejoindre l'iso-densité cible. Inversement, si la surface est localement à l'extérieur de la cible, elle devra se dégonfler pour aller épouser le bord. Cette façon de gonfler et dégonfler fait bien sûr penser à une bulle ou à un ballon (figure 1). Le concept est d'ailleurs très proche des "balloon snakes" de Laurent Cohen en analyse d'images [Coh91].

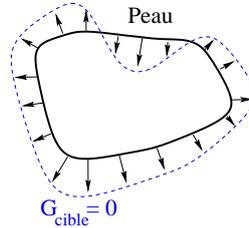


FIG. 1 - *Stratégie de la bulle : gonflements et dégonflements locaux*

3.4.2 Prise en compte d'une tension de surface

La stratégie peut aussi conférer à la peau une simulation de tension de surface. Par analogie à une peau élastique sous tension, il suffit d'ajouter un terme de pénalisation de la courbure pour simuler finalement une tension de surface. La stratégie devient alors:

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= \frac{d\mathbf{X}}{dt}(t)_{cible} + \frac{d\mathbf{X}}{dt}(t)_{tension} \\ &= \alpha (G_{cible}(\mathbf{X}) - iso) \mathbf{n}(\mathbf{X}) + \beta \kappa(\mathbf{X}) \mathbf{n}(\mathbf{X}) \\ &= [\alpha (G_{cible}(\mathbf{X}) - iso) + \beta \kappa(\mathbf{X})] \mathbf{n}(\mathbf{X}) \end{aligned} \quad (3)$$

où κ est la courbure moyenne, et β un coefficient permettant de régler l'importance de la tension de surface. Par cette dernière équation, on impose à la peau à la fois de poursuivre l'iso-densité et de minimiser sa courbure en tout point.

Cette stratégie de mouvement que nous avons défini est valide pour tout potentiel G_{cible} . Il peut même être simplement C^0 , puisque la peau lissera le contour grâce à la tension de surface. Ce potentiel cible peut bien sûr changer au cours du temps, comme cela sera le cas pour l'habillage de particules.

4 Implémentation

Les éléments exposés au paragraphe précédent peuvent difficilement s’implémenter directement. Une méthode complète et précise pour intégrer les équations différentielles de type Hamilton-Jacobi comme l’équation (1) a été mis au point par Sethian [Set96]. Bien que cette théorie est maintenant très élaborée, nous avons développé une autre approche offrant une solution efficace à ce problème.

4.1 Considérations simplificatrices

Pour réduire le nombre de calculs, il est important de se débarrasser de tout le superflu pour ne garder que l’essentiel. La peau n’ayant qu’un rôle surfacique, il est logique de limiter le calcul du potentiel discret f à une zone proche de la surface.

4.1.1 Voisinage Tubulaire

Comme Velho et Gomes l’ont fait remarquer [VG96], les caractéristiques fondamentales d’un potentiel sont ses variations autour de la surface implicite qu’il définit. En dehors de ce voisinage tubulaire de la surface, les valeurs du potentiel n’ont aucune influence sur la surface elle-même.

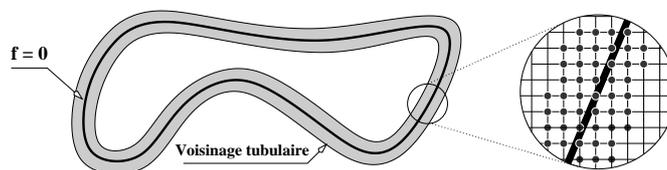


FIG. 2 - Voisinage tubulaire d’une surface implicite définie par $f = 0$.

Dans notre cas, où le potentiel est discret, ce voisinage se traduit par un ensemble de points de la grille tout autour de la surface (figure 2). L’information contenue dans le voisinage tubulaire permet ainsi de déduire le gradient du potentiel sur la surface par différences finies. On s’aperçoit qu’il n’est donc pas nécessaire d’intégrer les variations de f sur toute la grille : il suffit de *propager un front de quelques voxels d’épaisseur* pour bénéficier de toutes les informations nécessaires à l’évolution de la surface.

4.1.2 Nœuds autour de la surface

Restreindre les calculs à un voisinage proche de la surface va permettre à tout instant d’optimiser à la fois les opérations d’intégration du champ et celles nécessaires à sa polygonalisation. En limitant les calculs au voisinage strictement nécessaire, on gagne non seulement en efficacité, mais de plus, *le coût de l’animation de la peau devient proportionnel à la surface couverte*, et non au nombre total de nœuds de discrétisation de f à l’intérieur du volume qui nous intéresse, qui est $O(n^3)$ pour une grille de $n \times n \times n$. En pratique, cela fait gagner un ordre de grandeur et ramène la complexité moyenne à $O(n^2)$.

4.1.3 Seuillages intérieur et extérieur

Les régions hors du voisinage tubulaire de la surface ne sont d’aucun intérêt. Il est pourtant bon de savoir si elles sont internes, ou externes à l’objet : cela permettra une détection des collisions efficace. On seuille donc le potentiel comme le propose Ross Whitaker [Whi95] : tous les nœuds dont la valeur de f dépasse 1 sont seuillés à 1, et donc considérés comme intérieurs. Inversement, tous les nœuds dont la valeur passe en dessous de -1 sont seuillés à -1 et considérés comme extérieurs. Le choix de la valeur seuil est bien sûr arbitraire : toute autre valeur changerait l’échelle du potentiel discret, mais pas la surface.

4.2 Structure de données

4.2.1 Codage du potentiel

Le potentiel étant un ensemble de valeurs discrètes sur une grille 3D, nous utilisons pour le stocker un tableau $n \times n \times n$ de nombres réels, où n sera fixé en fonction de la finesse de polygonalisation

souhaitée [FV97].

De plus, le seuillage entre -1 et 1 , décrit au paragraphe 4.1, peut conduire à une petite optimisation : les valeurs de f peuvent être codées en virgule fixe, sur par exemple 2 octets, assurant une précision de $1/32768$ pour un gain de place mémoire d'un facteur 4 par rapport à un codage en double precision.

4.2.2 Liste chaînée des nœuds autour de la surface

De façon à pouvoir accéder rapidement aux seuls nœuds intéressants, à savoir ceux qui se trouvent dans un voisinage tubulaire de la surface, nous utilisons une simple liste chaînée comprenant les coordonnées de ces nœuds dans le tableau du potentiel. En effet, cette liste peut être tenue à jour dynamiquement de façon très rapide par suppressions et additions de nœuds au cours du mouvement de la surface. Une structure de données plus complexe n'est pas nécessaire : l'ordre de stockage des nœuds n'a aucune importance, puisque le traitement de chaque nœud est indépendant. Ainsi, nous conservons les coordonnées des nœuds *utiles* au cours du mouvement de la surface, et l'on verra par la suite que cette structure suffit.

4.3 Intégration robuste

La stratégie de vitesse instantanée exposée au paragraphe 3.4 n'était définie que pour des points sur la surface de la peau. Or l'intégration se fait uniquement sur les nœuds de la grille proches de la surface : il s'agit donc d'étendre la notion de champ de vitesses au voisinage de la surface.

4.3.1 Problème d'oscillation

Une intégration naïve de l'équation différentielle sur tous les points de la grille voisins de la surface amène d'inévitables oscillations. Pour s'en convaincre, il suffit de comprendre que la vitesse d'un nœud ne peut être nulle que si ce nœud est juste sur la surface cible fixée : alors, $G_{cible}(\mathbf{X}) = iso$ et $\frac{d\mathbf{X}}{dt}$ est nul. Les nœuds étant sur une grille régulière, ce cas de figure est évidemment très rare : cela signifie que le système n'arrivera jamais à un état de repos, oscillant toujours autour de la position cible.

En effet, l'équation différentielle (1) n'a de sens que pour des points *sur la surface*. Pour remédier à cet inconvénient, il est judicieux de formuler la stratégie aux sommets de la grille *en fonction du point le plus proche de la surface actuelle*. Ainsi, si la peau passe exactement par la surface cible, les sommets voisins seront bien affectés d'une vitesse nulle. Cela équivaut à *faire bouger les isopotentielles voisines à la même vitesse que la surface*, comme le représente la figure 3 : c'est d'ailleurs la méthode proposée, mais non utilisée en pratique, par Malladi et Sethian [MSV95], car cette solution amène un nouveau problème : comment trouver rapidement le point le plus proche ?

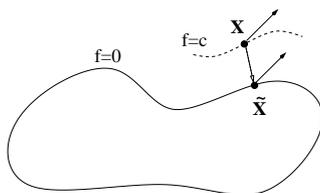


FIG. 3 - Utilisation du point de la surface le plus proche (d'après [MSV95])

4.3.2 Évaluations sur le maillage polygonal

Nous proposons, plutôt que de réellement déterminer la projection $\tilde{\mathbf{X}}$ sur la surface actuelle d'un point \mathbf{X} de la grille, d'approximer $G_{cible}(\tilde{\mathbf{X}})$ grâce aux informations locales dont on dispose déjà. Si, lors de la dernière polygonalisation de la peau, des voxels autour de \mathbf{X} étaient à cheval sur la surface, un ensemble de p points $\{\mathbf{P}_i\}$ sur la surface actuelle ont été déterminés par interpolation linéaire¹. Alors, nous remplaçons tout simplement l'évaluation de $G_{cible}(\tilde{\mathbf{X}})$ par :

$$\frac{1}{p} \sum_{i=1..p} G_{cible}(\mathbf{P}_i).$$

1. Ce cas est relativement courant, puisque l'on limite tous les calculs au voisinage tubulaire de la surface.

Ceci peut s'implémenter fort simplement en examinant si, dans les six arêtes partant de \mathbf{X} , il y a des points trouvés comme appartenant à la surface lors la polygonalisation précédente. Cela nécessitera donc jusqu'à six évaluations du potentiel cible par sommets de la grille, mais un système d'estampillage évitera de refaire des évaluations lors du traitement des nœuds voisins. Cette méthode nécessite du coup autant d'évaluations de G_{cible} que de points d'échantillonnage, évitant ainsi une surcharge de calculs. Dernier cas : si le nœud \mathbf{X} ne possède aucun point sur la surface dans son voisinage, on se contente de la classique évaluation de $G_{cible}(\mathbf{X})$ sans que cela ne prête à conséquence : \mathbf{X} est "assez" loin de la surface.

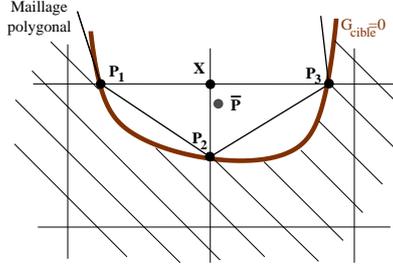


FIG. 4 - Evaluation de la stratégie en utilisant les points de la surface

Par cette méthode, nous sommes alors assurés de la bonne intégration du mouvement, et surtout, d'un état d'équilibre stable une fois que les polygones passent bien par la surface cible. Et cela sans recourir à une viscosité artificielle, mais simplement en décalant l'évaluation du potentiel cible, introduisant un filtrage suffisant pour nos besoins.

On remarque qu'ainsi, à chaque pas de temps, on se sert de *tous* les points de discrétisation courants pour définir comment le potentiel (et par effet de bord, la surface) doit exactement évoluer.

4.4 Ajout de la tension de surface

Puisque simuler une tension de surface revient à minimiser la courbure de la peau en tout point, il faut pouvoir exprimer la courbure moyenne locale en tout point, à savoir la moyenne des deux courbures principales. La surface est définie implicitement par le potentiel f , il est possible alors d'écrire la courbure moyenne comme suit :

$$\begin{aligned} \kappa(\mathbf{X}) &= \text{div}(\mathbf{n}(\mathbf{X})) \\ &= \text{div}\left(-\frac{\nabla f(\mathbf{X})}{\|\nabla f(\mathbf{X})\|}\right) \\ &= -\frac{f_x^2(f_{yy}+f_{zz})+f_y^2(f_{xx}+f_{yy})+f_z^2(f_{xx}+f_{yy})-2f_x f_y f_{xy}-2f_y f_z f_{yz}-2f_x f_z f_{xz}}{(f_x^2+f_y^2+f_z^2)^{3/2}} \end{aligned} \quad (4)$$

en fonction des différentes dérivées partielles (f_x, f_{xx}, \dots) approximées en \mathbf{X} par différences finies.

Les courbures étant connues aux sommets de la grille, on peut tout simplement calculer la composante $\beta\kappa(\mathbf{X})$ en chacun de ces sommets. En effet, il s'avère en pratique qu'une telle évaluation suffit à une bonne animation de la peau, sans recourir à des évaluations plus fines comme c'est obligatoirement le cas pour la composante $\alpha(G_{cible} - iso)$.

4.5 Mise à jour du voisinage tubulaire

Un point important à aborder est la gestion dynamique des nœuds du voisinage tubulaire de la surface. Il s'agit en effet de savoir quand ajouter un nœud dans le voisinage, et quand doit-il être enlevé.

Le "réveil" d'un nœud doit tout simplement avoir lieu quand la surface arrive à proximité immédiate de l'un de ses voisins. Une mise en œuvre simple peut donc se faire lors du calcul de la vitesse $d\mathbf{X}/dt$ sur un nœud : si la surface passe dans un des voxels adjacents, on vérifie que tous nos voisins sont déjà actifs, et, au besoin, on réveille ceux qui ne le sont pas. Autrement dit, on propage l'information à ses proches voisins : par ce principe de "bouche à oreille", on anticipera l'évolution de la surface en activant les nœuds quand nécessaire.

Quant à la suppression d'un nœud, elle s'effectue lors du seuillage : les nœuds de la grille sortant de l'intervalle $] - 1, 1[$ sont seuillés et sortis de la liste chaînée, car ils ne font plus partie du voisinage

tubulaire de la surface.

Par ces principes très simples, on tient à jour la liste des nœuds dans le voisinage tubulaire, avec un surcoût minime de calcul.

4.6 Construction du maillage et de ses normales

Comme on l’a vu précédemment, l’affichage de la peau à la fin de chaque pas de temps peut se faire de façon simple et rapide. En parcourant les voxels compris dans le voisinage tubulaire de la surface, on génère des polygones en traitant chacun des voxels selon les valeurs du potentiel aux sommets. On construit ainsi une liste de points d’échantillonnage et une liste de facettes s’appuyant sur ces points.

Cependant, pour permettre un rendu agréable de type “shading²”, il faut aussi associer à chaque point une normale. Comme l’on dispose par différences finies des normales à la surface, il suffit de les interpoler pour trouver la normale à chaque point du maillage [LC87]. On dispose ainsi de toutes les données nécessaires pour soit faire un rendu immédiat, soit stocker les informations en vue d’y appliquer un lancer de rayons pour améliorer encore la qualité des images.

Appliqué à des surfaces cibles très lisses, l’algorithme décrit ci-dessus exhibe malheureusement une sorte d’*acné de surface* quand la tension de surface est nulle : autant la polygonalisation est irréprochable, autant le shading crée des effets de petites bosses sur toute la surface (bump mapping). C’est en effet un défaut qui est dû à des petites variations de la normale par rapport à la véritable normale : on reconstruit bien la surface sur la grille, mais les normales ne sont qu’approximées par différences finies. Cet inconvénient peut être réglé par un simple *lissage*. Et ce lissage est justement obtenu par une tension de surface, même très faible. En conséquence, la peau générée par le modèle muni d’une tension de surface apparaît parfaitement lisse, sans avoir recours à un traitement particulier des normales.

4.7 Algorithme global

La mise en œuvre des points détaillés ci-dessus peut se faire en deux passes : il faut en effet faire l’évaluation des vitesses sans toucher aux valeurs de f dans un premier temps, puis, lors d’une deuxième passe, les mettre à jour en intégrant. Ce qui résulte en l’algorithme suivant :

À chaque pas de temps dt ,

- Pour tous les points \mathbf{X} de la grille référencés par la liste chaînée (*voisinage tubulaire*)
- Calculer $\nabla f(\mathbf{X}, t)$ par différences finies.
- Calculer $\frac{d\mathbf{X}}{dt}$ comme décrit aux sections 4.3.2 et 4.4.
- En déduire $\frac{\partial f}{\partial t}(\mathbf{X}, t) = -\nabla f(\mathbf{X}, t) \cdot \frac{d\mathbf{X}}{dt}$
- Pour ces mêmes points \mathbf{X}
- Mettre à jour : $f(\mathbf{X}, t + dt) = f(\mathbf{X}, t) + \frac{\partial f}{\partial t}(\mathbf{X}, t) dt$
- Visualiser la nouvelle surface en polygonalisant les voxels du voisinage tubulaire.

4.8 Pas de temps

Le dernier point important à aborder est le problème du choix du pas de temps pour l’intégration, afin de disposer finalement d’une méthode stable et précise.

4.8.1 Cas de la stratégie sans tension de surface

L’équation différentielle (1) est de type purement hyperbolique dans le cas où il n’y a pas de tension de surface, c’est-à-dire si $d\mathbf{X}/dt = (d\mathbf{X}/dt)_{cible}$. Son pas de temps doit donc suivre le classique critère de Courant-Friedrichs-Levy, à savoir, dans notre cas :

$$dt \leq \frac{\Delta x}{v} \tag{5}$$

2. Rendu de la surface tenant compte des normales pour le calcul de l’éclairage.

où $v = \max \left\| \frac{d\mathbf{X}_i}{dt} \right\|$ représente la vitesse maximale courante des nœuds. Ce critère est en tout point analogue au théorème de Shannon, qui stipule qu’il faut échantillonner le temps au double de la fréquence la plus élevée d’un signal pour ne pas perdre en qualité.

Au vu de l’algorithme que nous utilisons, il faut aussi vérifier que la surface ne puisse “traverser” plus d’un nœud en un pas de temps : sinon, notre liste chaînée stockant les nœuds du voisinage tubulaire risquerait de n’être plus valide. Il faut et il suffit, pour éviter cela, que $v \cdot dt$ soit inférieur à Δx : on retrouve exactement le critère de Courant. Ce n’est donc pas une contrainte supplémentaire.

4.8.2 Cas de la stratégie avec tension de surface

L’ajout d’une tension de surface modifie la taille du pas de temps à utiliser. Le terme de courbure change en effet le type d’équation différentielle : on rajoute une contribution *parabolique* à une équation hyperbolique. La contrainte sur le pas de temps devient alors :

$$dt \leq \frac{\Delta x^2}{2D}$$

où Δx représente la taille de la discrétisation spatiale, et $D = \beta \max \|\nabla f(\mathbf{X}_i)\|$ représente la diffusion due à la tension de surface.

Finalement, le pas de temps choisi dans notre implémentation s’écrit :

$$dt = \min\left(\frac{\Delta x}{v}, \frac{\Delta x^2}{2D}\right) \quad (6)$$

5 Extensions des propriétés de la peau

Le modèle de base étant défini, il est possible de l’enrichir de nouvelles propriétés physiques, sans que l’algorithme ne change. Ces propriétés, comme la surface ou le volume constant, étaient délicates à mettre en œuvre dans les modèles implicites précédents [DG95] : elles trouvent pourtant ici une solution simple.

5.1 Surface constante

Lors de la polygonalisation, il est aisé de sommer la surface de chaque triangle généré afin d’obtenir une approximation de l’aire totale de la surface. On peut alors, à l’instar de la tension de surface, rajouter une pénalisation qui permettra par exemple de conserver une aire donnée. Il suffit pour cela d’ajouter le terme :

$$\frac{d\mathbf{X}}{dt}(t)_{surface} = \gamma \frac{(A_0 - A(t))}{A_0} \mathbf{n}(\mathbf{X})$$

à la stratégie de champ de vitesses. Cela fera enfler ou déenfler globalement l’objet qui gardera une surface constante, ou qui peut, sans plus de difficultés, suivre une évolution donnée si l’on fait varier A_0 au cours du temps.

5.2 Volume constant

Le volume total approximé par la représentation polygonale peut être tenu à jour très simplement aussi. À chaque changement de signe d’un nœud, il suffit d’oter ou d’ajouter le volume d’un voxel pour disposer d’une approximation suffisante du volume. Une implémentation plus approfondie peut calculer le volume finement en tenant compte de la polygonalisation de chaque cube. Dans les deux cas, il suffit d’ajouter le terme de pénalisation :

$$\frac{d\mathbf{X}}{dt}(t)_{volume} = \gamma \frac{(V_0 - V(t))}{V_0} \mathbf{n}(\mathbf{X})$$

pour que la stratégie tienne également compte du contrôle de volume.

Il est intéressant de remarquer ici que cette méthode de conservation de volume est très similaire à la technique proposée par Emmanuel Promayon [Pro97]. Il a en effet prouvé que le gradient de la contrainte d’incompressibilité d’un objet maillé en un sommet du maillage est colinéaire à la moyenne des normales

des facettes adjacentes. C'est-à-dire, dans notre cas, que la direction maximisant la variation de volume est *celle de la normale à notre peau*: le choix du déplacement selon la normale, fait au paragraphe 3.4, est donc justifié *a posteriori* dans ce cas.

6 Résultats

Cette peau active a bien entendu été testée dans une optique d'habillage de modèles physiques, mais d'autres applications ont été envisagées.

6.1 Habillage de particules

Notre modèle de peau peut être incorporé à un système de particules. Les premiers essais ont été faits avec des systèmes de particules conventionnels. Ces particules servant de squelettes à une surface implicite définissant la chair, il est alors simple de faire converger la peau vers cette surface implicite qui évolue en fonction du mouvement des particules.

Dans cet exemple, nous avons simulé la peau à partir des positions de particules obtenues lors d'une simulation simple. Il n'y a donc pas d'interaction entre peau et particules. On voit cependant sur la figure 5 que la peau enrobe les particules et dissimule ainsi la granularité du modèle. On remarque aussi que la tension de surface pourra permettre d'obtenir divers aspects pour une même animation de particules. Les mêmes tests peuvent être effectués avec des particules adaptatives cette fois, ce qui permet de masquer les changements de granularité au sein du matériau.

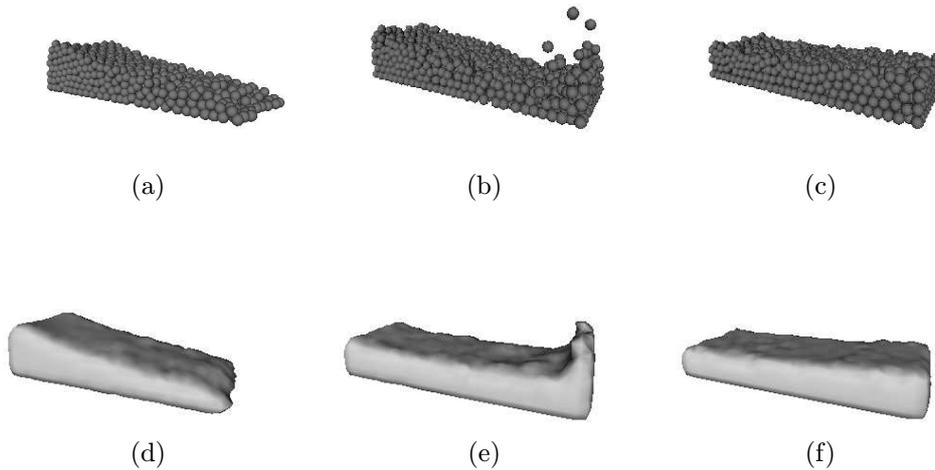


FIG. 5 - Exemples d'habillage de particules, pour $\beta = .15$, avec une taille de grille de $40 \times 20 \times 20$.

Pour être plus réaliste, il faudra encore dans un proche avenir créer une véritable rétroaction entre peau et système de particules, comme s'était le cas dans notre modèle précédent [CGD97]. Cela semble tout à fait possible, lors des évaluations du potentiel cible, grâce aux points d'échantillonnage. Nous n'avons pas encore développé cette dernière mise au point.

6.2 Métamorphose automatique

Une application directe de la technique décrite dans ce papier est le *morphing* (métamorphose) automatique entre objets implicites. En effet, si la peau est initialisée sur une forme de départ, et que l'on change la définition de sa surface cible, la peau se met à évoluer de façon *autonome* vers la nouvelle forme.

La spécificité d'une telle méthode vis-à-vis des techniques classiques de métamorphose entre objets implicites définis par squelettes est qu'il n'est absolument pas nécessaire de faire une phase de correspondance entre squelettes [Gal97]. La peau se transforme d'une forme à une autre sans aucune indication. C'est bien sûr aussi un défaut : on ne pourra pas contrôler comment la métamorphose doit se faire. Seuls les variations des potentiels et les paramètres α et β pourront être modifiés pour changer le comportement obtenu.

La figure 6 montre une métamorphose entre deux objets de même topologie. L'objet de départ est la composition d'un vissage et d'un potentiel défini par un segment, alors que l'objet cible est un potentiel analytique. Les paramètres utilisés sont respectivement $\alpha = 2.0$, $\beta = .01$, et le potentiel discret est stocké sur une grille de taille $50 \times 50 \times 50$ pour obtenir une représentation très fine. Le temps de calcul moyen entre deux images est inférieur à la seconde sur une *Indy R4600*.

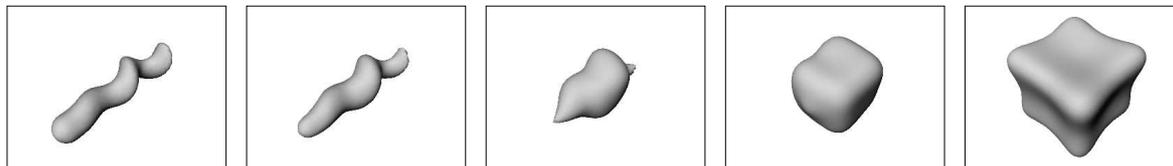


FIG. 6 - Exemple de métamorphose entre deux objets de même topologie.

Le deuxième exemple illustré par la figure 7 montre une métamorphose entre deux objets de topologies différentes. L'objet de départ est un mélange de six squelettes ponctuels, et l'objet cible est une paire de sphères, les paramètres restant inchangés par rapport à l'exemple précédent. La configuration des deux objets implicites offre ici un effet intéressant : quatre des six lobes de l'objet initial semblent caviter, alors que les deux derniers lobes paraissent s'écarter sur les côtés. Les surfaces intermédiaires exhibent alors des arêtes presque vives. Le filament joignant les deux sphères disparaît peu à peu, pour finalement donner lieu à une séparation.

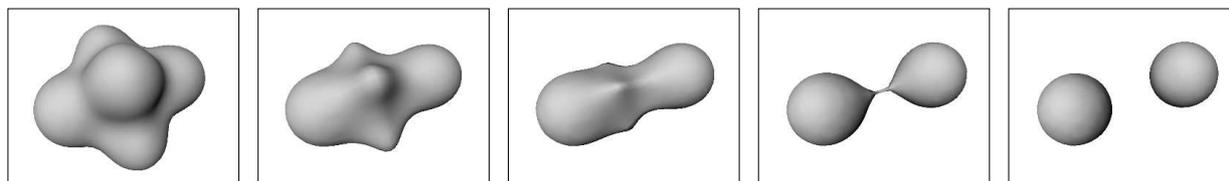


FIG. 7 - Exemple de métamorphose entre objets de topologies différentes

7 Conclusion et perspectives

Cet article a présenté une nouvelle technique de simulation de peau. Le modèle utilise une formulation implicite discrète, apte à gérer tout changement de topologie. À partir des points d'échantillonnage trouvés au pas précédent, le potentiel discret définissant la peau est modifié en fonction du champ de vitesses imposé, provoquant donc de façon indirecte le mouvement souhaité pour la peau. Ce comportement, déterminé par le champ de vitesses à chaque instant, peut être défini par plusieurs stratégies. Nous avons choisi, dans le cadre de l'animation de matériaux déformables, d'attirer la peau vers la chair tout en simulant une tension de surface. Les paramètres d'attraction (α) et de tension (β) étant macroscopiques, leurs effets sur le mouvement sont très intuitifs.

Cette méthode rapide, car réduisant ses calculs à une proche périphérie de la surface, permet une gestion de surfaces implicites mouvantes de complexité variable sans surcoût de calcul contrairement aux méthodes à base de squelettes. De plus, le temps de détection de collision est encore réduit grâce à l'emploi d'un potentiel simplissime. Enfin, la surface générée pouvant être munie de propriétés physiques supplémentaire comme une surface ou un volume constant, son usage permet d'espérer une gamme importante d'applications.

Utilisée en tant qu'enrobage d'un système de particules, cette peau assure un filtrage temporel intéressant. Selon la tension de surface et la vitesse d'attraction définies, un mouvement pré-calculé pourra être grandement filtré par la simulation du mouvement de la peau.

Une autre utilisation immédiate de cette peau est l'habillage de modèles Euleriens [FM96]. En effet, dans la description Eulerienne, le champ de vitesses du matériau est déjà géré : la peau pourra ainsi suivre les mouvements générés de façon aussi simple qu'avec les particules.

Enfin, on peut songer à ajouter une détection des changements de topologie, qui peut être utile à la gestion de la chair. Bien qu'un parcours du tableau des valeurs du potentiel puisse permettre de trouver le nombre de composantes connexes, une méthode simple comme la caractéristique d'Euler-Poincaré

permet, à partir du nombre courant de triangles, d'arêtes et de sommets dans la polygonalisation de la peau, de déterminer les changements de topologie, comme utilisé dans [LM96].

Références

- [CGD97] Marie-Paule Cani-Gascuel and Mathieu Desbrun. Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):39–50, March 1997.
- [Coh91] Laurent D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing : Image Understanding*, 53(2):211–218, March 1991.
- [Des97] Mathieu Desbrun. Modélisation et animation de matériaux hautement déformables. *Thèse de docteur ingénieur en informatique*, Institut National Polytechnique de Grenoble, December 1997.
- [DG95] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 287–290. ACM SIGGRAPH, Addison Wesley, August 1995. Los Angeles, CA.
- [FM96] Nick Foster and Dimitri Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5):471–483, 1996.
- [FV97] Max Froumentin and Eric Varlet. Dynamic implicit surface tessellation. In *ACM Symposium on Virtual Reality Software and Technology'97 Proceedings*, Lausanne, Switzerland, 1997.
- [Gal97] Eric Galin. Métamorphose et visualisation de blobs à squelettes. *Thèse de Doctorat*, Université Claude Bernard, Lyon, July 1997.
- [GVP91] Marie-Paule Gascuel, Anne Verroust, and Claude Puech. A modeling system for complex deformable bodies suited to animation and collision processing. *Journal of Visualization and Computer Animation*, 2(3):82–91, August 1991. A shorter version of this paper appeared in *Graphics Interface'91*.
- [Hab97] Arash Habibi. Visualisation d'objets très déformables : relations mouvement - forme - image. *Thèse de Doctorat*, INPG, January 1997.
- [LB94] Jacques-Olivier Lachaud and Eric Bainville. A discrete adaptive model following topological modifications of volumes. In *Discrete Geometry for Computer Imagery*, pages 183–194, September 1994.
- [LC87] William Lorensen and Harvey Cline. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California, July 1987).
- [LHVD95] Annie Luciani, Arash Habibi, André Vapillon, and Yves Duroc. A physical model of turbulent fluids. In *6th Eurographics Workshop on Animation and Simulation*, Maastricht, Netherlands, September 1995.
- [LJR⁺91] Annie Luciani, Stéphane Jimenez, Olivier Raoult, Claude Cadoz, and Jean-Loup Florens. A unified view of multitude behaviour, flexibility, plasticity, and fractures: balls, bubbles and agglomerates. In *IFIP WG 5.10 Working Conference*, Tokyo, Japan, April 1991.
- [LM96] Jacques-Olivier Lachaud and Annick Montanvert. Volumic segmentation using hierarchical representation and triangulated surface. In *European Conference on Computer Vision*, pages 137–146, Cambridge, U.K., April 1996.
- [MSV95] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.
- [Pro97] Emmanuel Promayon. Modélisation et simulation de la respiration. *Thèse de Doctorat*, November 1997.
- [Set96] James A. Sethian. *Level Set Methods*. Cambridge Press, 1996.
- [SS96] Jean-Paul Smets-Solanes. Vector field based texture mapping of animated implicit objects. In *Eurographics'96*, pages 289–300, Poitiers, France, August 1996.
- [Ton91] David Tonnesen. Modeling liquids and solids using thermal particles. In *Graphics Interface'91*, pages 255–262, Calgary, AL, June 1991.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205–214, July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California).
- [TPF89] Demetri Terzopoulos, John Platt, and Kurt Fleisher. Heating and melting deformable models (from goop to glop). In *Graphics Interface'89*, pages 219–226, London, Ontario, June 1989.
- [Tur95] Russel Turner. Leman: A system for construsting and animating layered elastic characters. In *Computer Graphics- Developments in Virtual Environments*, pages 185–203, Academic Press, San Diego, CA, June 1995.
- [VG96] Luiz Velho and Jonas Gomez. Approximate conversion of parametric to implicit surfaces. *Computer Graphics Forum*, 15(5):327–337, December 1996. A preliminary version of this paper appeared in *Implicit Surfaces'95*, Grenoble, France, may 1995.
- [WH94] Andrew Witkin and Paul Heckbert. Using particles to sample and control implicit surfaces. *Computer Graphics*, pages 269–278, July 1994. Proceedings of SIGGRAPH'94.
- [Whi95] Ross Whitaker. Algorithms for implicit deformable models. In *The International Conference of Computer Vision*, Boston, Mass, 1995.
- [WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, August 1986.