



HAL
open science

Animation d'objets déformables à l'aide de surfaces implicites

Mathieu Desbrun, Marie-Paule Cani

► **To cite this version:**

Mathieu Desbrun, Marie-Paule Cani. Animation d'objets déformables à l'aide de surfaces implicites. journées du groupe de travail Animation et Simulation (GTAS), Jan 1996, Strasbourg, France. inria-00537535

HAL Id: inria-00537535

<https://inria.hal.science/inria-00537535>

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.

Animation d'Objets Déformables à l'aide de Surfaces Implicites

Mathieu Desbrun
Marie-Paule Gascuel

iMAGIS*/ IMAG
BP 53, F-38041 Grenoble cedex 09, France
Mathieu.Desbrun@imag.fr

Résumé

Cet article présente une approche générale pour la construction et l'animation de modèles déformables complexes. Les modèles physiques existants sont combinés à une couche élastique utilisant une surface implicite, ce qui apporte une solution élégante à certaines de leurs limitations. La formulation implicite définit en effet une surface lisse pour chaque objet, où la détection des collisions et la modélisation fine des déformations sont particulièrement efficaces. De plus, cette approche permet un contrôle fin des variations de volume au cours des déformations, même lorsque ces dernières sont accompagnées de changements de topologie. Les applications sont extrêmement variées, de la simulation d'objets munis de squelettes articulés à l'animation de substances hautement déformables.

1 Introduction

Dans les systèmes d'animation traditionnels, les mouvements et les déformations éventuelles des différents objets mis en scène devaient être entièrement spécifiés par un animateur. De plus en plus, des modèles physiques d'objets viennent simplifier cette lourde tâche. Il suffit maintenant de définir les conditions initiales et une loi comportementale pour engendrer le mouvement et les déformations d'un objet, en détectant et en traitant les collisions éventuelles. Ces modèles générateurs sont particulièrement utiles pour des objets déformables, qui autrement nécessiteraient un traitement lourd et très intuitif de la part de l'animateur. Un certain nombre de difficultés subsistent cependant lorsque des modèles déformables traditionnels (éléments finis, systèmes masses-ressorts ou à particules) sont utilisés, comme la définition d'une surface lisse autour des objets, la gestion des interactions et en particulier des situations de contact, ou le contrôle des variations de volume en cours de déformation.

Travaux antérieurs

La plupart des modèles physiques déformables sont basés sur une approche *nodale*, c'est-à-dire modélisant les déformations d'un objet par des déplacements de noeuds élémentaires à l'intérieur de ce dernier. C'est le cas des modèles de Demetri Terzopoulos [TPBF87, TF88] qui discrétisent une équation différentielle issue de la théorie mécanique des milieux continus. En échantillonnant l'espace et le temps, on peut en effet approximer par différences finies les dérivées des paramètres du mouvement sur les noeuds d'un maillage, et donc résoudre le mouvement de façon globale en résolvant un système matriciel. D'autres approches nodales [MP89, LJF⁺91] considèrent les objets déformables comme un assemblage d'éléments mécaniques discrets. Les noeuds sont alors des masses ponctuelles, reliées entre elles par des lois d'interaction de type ressort ou amortissement. Ce choix permet une

*. iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.

résolution indépendante sur chaque masse, en intégrant le mouvement à partir du bilan des forces.

Un aspect important d'un modèle physique est sa façon de traiter les collisions. Or la plupart des modèles que nous venons de citer souffrent des mêmes défauts dans ce domaine. Il y a bien souvent absence de cohérence entre surface affichée et surface utilisée pour la détection des collisions. De plus, la complexité de cette détection est généralement quadratique en nombre de noeuds¹. Enfin, beaucoup de modèles calculent la réponse aux collisions par une méthode de pénalité [MW88], ce qui signifie que les collisions sont simulées à partir d'interpénétrations plutôt que par l'évaluation de forces de contact surfaciques.

Vers un modèle hybride

Pour pouvoir profiter des différentes qualités de diverses modélisations, quelques auteurs ont proposés l'utilisation de *modèles hybrides* [TW88, CHP89]. Par modèle hybride, on désigne la séparation d'un objet déformable en au moins deux composants distincts. On utilise généralement une partie interne, simulant les déformations à grande échelle, et une partie externe déformable qui s'ajoute à la première pour définir la forme courante. L'utilisation d'un tel modèle hybride permet des temps de simulation raisonnables, et offre à l'utilisateur des paramètres indépendants sur le comportement physique macroscopique et les déformations locales, facilitant la mise au point des animations.

Dans la lignée des modèles hybrides, notre approche consiste à *enrober* une structure interne rigide ou déformable, basée sur un modèle existant, par une surface implicite. L'usage de ces surfaces permet d'habiller la structure interne indépendamment de sa topologie, ajoutant ainsi de l'attrait au modèle résultant. Il permet en même temps d'accélérer la détection de collisions, de modéliser finement les contacts et de gérer les variations de volume lors des déformations. Ce modèle hybride autorise des simulations très variées, y compris pour des objets capables de fractures ou de fusions.

Cet article s'attachera dans un premier temps à décrire l'enrobage implicite, puis nous détaillerons nos solutions aux problèmes de l'échantillonnage et du mélange de primitives posés par cette technique. Une fois la couche externe au point, nous décrirons comment contrôler efficacement son volume au cours des animations. Enfin, nous examinerons deux exemples concrets de modèles hybrides, l'un pour l'animation d'objets ou de personnages munis de squelettes articulés, et l'autre pour la simulation de substances hautement déformables.

2 Enrobage implicite

Dans cette section, nous allons tout d'abord rappeler la définition et les avantages des surfaces implicites pour la modélisation. Puis, après avoir décrit un modèle élastique basé sur ces mêmes surfaces, nous présenterons notre modèle hybride proprement dit.

2.1 Surfaces implicites

Une surface implicite est définie comme l'ensemble des points vérifiant une équation :

$$S = \{P \in \mathbb{R}^3 \mid f(P) = c\}$$

où c est appelé *isovaleur*. Les points intérieurs à l'objet sont définis par la propriété :

$$f(P) \geq c \tag{1}$$

Trivialement, l'extérieur possède le même type de caractérisation : $f(P) < c$. On dispose ainsi d'une fonction dedans/dehors, facilitant le rendu d'images composées d'objets implicites. Les techniques de lancer de rayons ont en effet besoin de connaître l'intersection entre un rayon et un objet. La normale

1. Elle peut cependant être améliorée grâce à la gestion de structures de données dynamiques relativement complexes[LC92].

d'un point à la surface d'un objet implicite, définie par une fonction f , est calculée par le gradient de f (noté ∇f) en ce point.

Les premières approches utilisèrent principalement des polynômes pour f . Les objets ainsi définis étaient donc essentiellement des ellipsoïdes ou des super-quadriques. Les formes trop limitées et le manque de paramètres intuitifs ont rapidement amené une reformulation complète [WW89]. On peut en effet définir plus généralement des surfaces implicites isopotentielles, engendrées par des squelettes. Ces squelettes peuvent être n'importe quelle primitive géométrique pour laquelle on sait calculer la distance à un point. Chaque squelette crée dans l'espace un potentiel qui décroît avec la distance. On considère alors qu'à chaque point de l'espace est associé le potentiel résultant de la somme des potentiels créés par chacun des squelettes. Une définition des surfaces définies par un ensemble de squelettes s_i est alors :

$$S = \{P \in \mathbb{R}^3 \mid \sum_{s_i} f_i(r_i) = c\}$$

où r_i est la distance de P à s_i et f_i est une fonction décroissante.

On remarque que la caractérisation de l'intérieur de l'objet reste identique (voir équation 1), et que la construction d'objets même complexes est assez intuitive. Un squelette ponctuel donnera une sphère, un segment donnera un cylindre arrondi aux extrémités, etc (figure 1). L'utilisation de plusieurs squelettes fournira un moyen simple et économique de création d'objets complexes. L'objet est ainsi défini de façon très compacte.

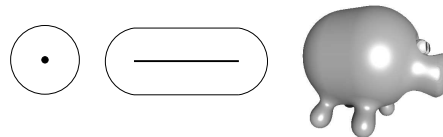


FIG. 1 - Exemples de surfaces implicites

Une fonction potentiel f_i peut être modélisée par toute fonction décroissante à support compact (voir figure 2). Le rayon d'influence R permet d'assurer qu'un squelette n'influera pas sur toute la forme d'un objet, mais seulement sur une zone locale, ce qui permet un contrôle local de la forme d'un objet complexe.

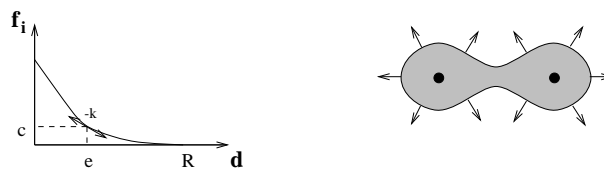


FIG. 2 - Une fonction potentiel et la surface engendrée par deux points-clé.

La variable e de f_i , que nous appellerons "épaisseur d'enrobage" représente le rayon qu'aurait la sphère engendrée par un squelette ponctuel muni de cette fonction. On a bien sûr la relation : $e = f^{-1}(c)$.

Les surfaces implicites ne servent pas qu'à la modélisation d'objets : un modèle [Gas93] a été proposé pour les adapter à la simulation d'objets élastiques. Notre enrobage implicite est basé sur ce modèle, que nous allons décrire dans le prochain paragraphe.

2.2 Modélisation des propriétés physiques d'un objet implicite

La forme d'un objet implicite ne dépend, comme on l'a vu, que de l'ensemble des points de l'isosurface. Les variations de la fonction potentiel autour de cette isosurface peuvent donc être exploitées pour modéliser d'autres propriétés. L'idée à la base de [Gas93] est d'utiliser ces variations pour stocker les propriétés physiques de l'objet. Ainsi, on augmente encore le taux de compression des informations sur l'objet.

Correspondance entre forces et déformations

Définir un modèle déformable, c'est décider d'une correspondance entre forces et déformations, qui détermine ainsi son comportement. Dans [Gas93], il est montré que si un objet implicite est défini par une fonction potentiel f vérifiant

$$\forall P, \nabla f(P) = -k(P).N(P) \quad (2)$$

où $k(P)$ représente la raideur du matériau au point P et $N(P)$ la normale, alors l'équivalence force/déformation s'écrit simplement:

$$g(P) = -N(P).R(P)$$

où g est le potentiel de déformation permettant aux deux objets en contacts de disposer d'une surface de contact précise (voir figure 3).

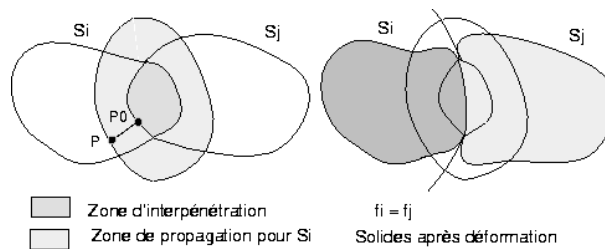


FIG. 3 - Modélisation fine du contact entre deux objets

Il est également montré qu'avec une expression très simple des potentiels de déformation (qui s'ajoutent respectivement aux potentiels f_i et f_j des deux objets en collision) :

$$g_i(P) = c - f_j(P)$$

$$g_j(P) = c - f_i(P)$$

on obtient une surface de contact précise entre objet (voir figure 3). Combinés avec l'équation d'équivalence entre force et déformation, on obtient alors des forces résultantes vérifiant bien la loi d'action-réaction. Cette méthode permet à la fois de calculer une déformation précise engendrant une vraie surface de contact et d'en déduire les forces de réaction résultantes.

Modéliser la raideur avec les fonctions implicites

Ce modèle permet aussi, en jouant sur la dérivé en c de la fonction potentiel, de modifier la raideur des objets. Ainsi, il est très facile pour un graphiste de choisir la raideur d'un objet : il suffit de changer la pente de la courbe au point e (vérifiant $f(e) = c$). De plus, la pente en tout point de la portion de courbe sur $[0, e]$ définit le comportement élastique à l'intérieur de l'objet. Deux choix sont souvent faits : soit on suppose que le matériau possède une élasticité linéaire (dans ce cas, la force de réaction est constante lors d'une déformation - voir figure 2), soit on préfère appliquer une élasticité quadratique; dans ce cas, la raideur augmente au fur et à mesure de la compression.

2.3 Mise au point d'un modèle hybride

L'utilisation des surfaces implicites permet la modélisation d'objets de forme libre et de topologie arbitraire, dont elles offrent de surcroît une représentation particulièrement compacte. Le modèle que nous venons de présenter permet d'étendre leur usage à la simulation d'objets élastiques et à la gestion précise des contacts surfaciques. Ces arguments en font une couche externe idéale pour notre modèle hybride. Ce dernier est donc défini de la manière suivante :

1. une structure interne (rigide ou déformable) servant de composant de référence,
2. un enrobage implicite, dont les squelettes sont attachés à certains éléments de la structure interne.

On dispose ainsi d'un modèle profitant des avantages à la fois de la structure interne et de l'enrobage implicite. Grâce à la couche implicite, les objets disposent maintenant d'une définition simple de leur surface, permettant de faire l'interface entre la partie interne et le monde extérieur, c'est-à-dire de détecter et de traiter les collisions rapidement. L'expression implicite assure aussi une modélisation précise des situations de contact, qui peuvent être rendues sans artefact visuel.

Le modèle étant construit, il faut maintenant planifier les différentes étapes d'animation à mettre en oeuvre. A chaque pas de temps :

1. Intégrer les équations du mouvement pour le composant interne en tenant compte des forces et des moments qui lui sont appliqués.
2. Déplacer la couche implicite en conséquence.
3. Traiter les interactions comme suit :
 - Détecter les interpénétrations.
 - Déformer les composants implicites de manière à engendrer des surfaces de contact exactes entre les objets.
 - Intégrer les forces de réaction et de frottement sur ces surfaces. Réactualiser le bilan des forces en conséquence.
4. Afficher les objets tels qu'ils ont été déformés.

Cet algorithme est utilisé avec un pas d'intégration adaptatif. En effet, au cas où l'intervalle de temps est trop grand, une importante interpénétration pourrait créer des déformations, et donc des forces complètement anormales. Ce problème est détecté et le système revient alors en arrière avec un intervalle de temps plus petit.

3 Problèmes intrinsèques aux surfaces implicites

L'utilisation directe des concepts évoqués plus haut n'est pas satisfaisante. Différents problèmes liés à la combinaison d'une surface implicite et d'une structure interne déformable doivent être réglés. En premier lieu, travailler avec des surfaces implicites, sans formulation explicite, nous oblige à maintenir à tout instant de la simulation un bon échantillonnage de notre surface. Apparaît aussi le problème de mélange entre squelettes au cours des mouvements. Les paragraphes suivants exposent nos solutions pour résoudre les problèmes de l'échantillonnage adaptatif des surfaces et du mélange à distance, généralement indésirable.

3.1 Echantillonnage

L'échantillonnage des surfaces implicites est un point crucial : ces points d'échantillonnage sont nécessaires pour la détection des collisions et l'intégration des forces de contacts. Malheureusement, la majorité des techniques développées pour les surfaces implicites demandent un temps de calcul

important à chaque mouvement de la structure interne. En effet, les méthodes d'échantillonnage par partitionnement spatial [WMW86] ne profitent pas de la cohérence spatiale d'une animation, obligeant à recalculer la discrétisation à chaque pas de temps. Il est donc préférable de choisir une méthode adaptative, comme celle présentée récemment par Andrew Witkin [WH94], qui échantillonne la surface à l'aide de particules répulsives, assurant ainsi à la fois une bonne répartition et un temps de calcul réduit entre deux pas de temps consécutifs.

Notre approche, détaillée dans [DTG95], se distingue de cette dernière méthode sur principalement trois points :

- Tout d'abord, notre idée de base est que chaque squelette dispose de ses propres *graines*, ou points d'échantillonnage. Ces graines se déplacent sur des axes partant du squelette pour venir se “coller” à la surface implicite après une recherche dichotomique. On profite bien sûr énormément de la cohérence temporelle à chaque petit déplacement du squelette, chaque graine partant de sa position précédente dans le repère local de ce dernier.
- Le deuxième point de différence est la mise au point d'un critère d'invalidation des graines : pour non seulement éviter de perdre du temps dans l'échantillonnage, mais aussi pour respecter une bonne répartition des échantillons, on stoppe la migration des graines le long de leurs axes quand elles pénètrent dans une zone où le potentiel créé par leur squelette émetteur n'est plus prépondérant. Ce critère définit alors une zone de prépondérance propre à chaque squelette : ces zones sont appelées *territoires*. Formellement, le territoire T_i associé au squelette s_i est défini par :

$$T_i = \{P \in \mathbb{R}^3 \mid (f(P) \geq c) \text{ and } (\forall j \ f_i(P) \geq f_j(P))\} \quad (3)$$

- Enfin, cet échantillonnage permet une représentation polygonale de l'objet à faible coût, utilisée pour la visualisation interactive des animations.

3.2 Mélange indésirable

Le mélange systématique de toutes les primitives d'un objet entre elles apporte un inconvénient majeur : on ne peut pas, par exemple, modéliser un personnage avec les bras le long du corps (voir figure 4).

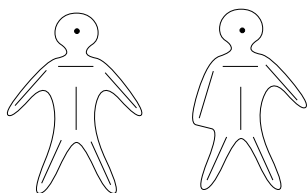


FIG. 4 - *Le problème du mélange indésirable*

Cas des objets ne changeant pas de topologie

Le problème a été résolu pour les objets ne changeant pas de topologie grâce à l'utilisation d'un graphe de mélange [WW89, OM93], définissant quelles primitives doivent se mélanger et lesquelles ne le doivent pas. L'algorithme est alors simple : pour un point P de l'espace, on trouve en premier lieu quel est le squelette prépondérant en P (c'est-à-dire celui de plus grand potentiel), et on somme les contributions de ce squelette et de ses voisins dans le graphe.

Cet algorithme convient à l'animation de personnages ; malheureusement, dans le cadre général qui est le nôtre, un objet est susceptible de changer de topologie, en se cassant par exemple.

Cas des objets pouvant subir des fractures

L'un des avantages des surfaces implicites est de pouvoir modéliser les fractures : un objet peut passer d'une seule composante connexe à plusieurs, du fait du mouvement relatif de ses squelettes. Cependant, si deux morceaux d'un objet qui s'est fracturé viennent à se rapprocher, ils se mélangent à nouveau, à distance. On comprend donc que l'on ne peut se satisfaire d'un graphe de mélange fixe pour un tel objet : il faut mettre au point un graphe de mélange *dynamique*.

Notre solution consiste à définir pour chaque squelette une "liste de mélange", c'est-à-dire la liste de tous les squelettes devant se mélanger avec lui. Ce graphe de mélange reste valide jusqu'à l'apparition d'une nouvelle composante connexe (fracture). On le maintient à jour au cours du temps par l'algorithme suivant, exécuté à chaque pas de l'animation :

- Pour chaque paire de squelettes, vérifier si leurs zones d'influence (définies par le rayon d'influence de leur potentiel) s'intersectent. La fermeture transitive de cette relation nous donne alors les différentes composantes connexes du *graphe d'influence* (voir figure 5).
- Pour chaque squelette, enlever de sa liste de mélange les squelettes n'étant plus dans la même composante connexe que lui.

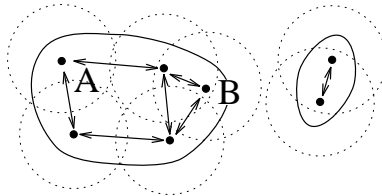


FIG. 5 - Le graphe d'influence et ses composantes connexes. Les squelettes A et B résident dans la même composante : leurs potentiels s'additionneront donc s'ils s'approchent l'un de l'autre.

Modification du traitement des collisions

Reste, une fois ce graphe défini, à traiter correctement les collisions, y compris les auto-collisions entre différentes parties d'un même objet. Pour cela, on détecte et on traite les collisions *entre chaque paire de territoires* associés à des squelettes ne se mélangeant pas au sens du graphe de mélange. Ceci assure, dans le cas de la figure 4, un traitement de collision entre le bras et le corps.

4 Contrôle du volume

Pouvoir contrôler le volume lors d'une animation est capital [PB88]. Certaines simulations physiques se doivent par exemple de conserver le volume au cours du mouvement. Pourtant, une surface implicite définie par des primitives simples voit son volume varier lorsque les primitives se déplacent les unes par rapport aux autres. Bien que souvent constaté [WW89], ce problème n'a jamais été abordé dans le cas général qui est le nôtre, où l'on peut avoir affaire à des fractures ou autres changements de topologie importants. Nous allons voir que grâce à notre méthode d'échantillonnage, ce problème trouve une solution élégante.

Détection des variations locales de volume

Contrôler le volume signifie en premier lieu savoir le déterminer. Or une résolution analytique n'est pas pensable dans ce cadre. Reste, parmi les solutions classiques, la discrétisation de l'espace en voxels. Cependant cette méthode souffre de deux problèmes différents : premièrement, elle est relativement coûteuse, et de plus, elle ne permet pas de déterminer les régions où le volume a changé. On ne disposerait que d'une valeur globale du volume.

Il semble souhaitable de pouvoir agir de façon locale sur le volume : comme le montre la figure 6, dans le cas d'un mouvement relatif entre squelettes, il est préférable de contrôler le volume localement près des squelettes concernés plutôt que d'effectuer une correction globale qui déformerait tout l'objet.

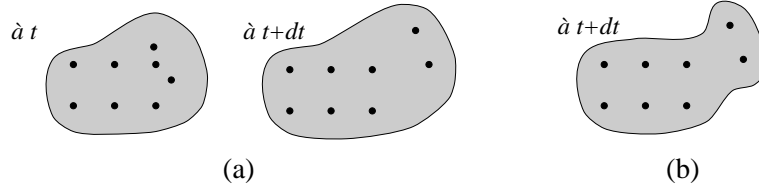


FIG. 6 - (a) Variation de volume. (b) Correction locale du volume.

C'est pourquoi nous avons repris l'idée des *territoires*, définis pour l'échantillonnage (voir équation 3). En effet grâce à ces territoires, chaque squelette dispose d'un volume (le volume de son territoire) qui lui est propre; l'ensemble de ces volumes forme une partition du volume global de l'objet. Approximer ce volume est chose aisée, encore une fois grâce à la méthode d'échantillonnage choisie: il suffit de sommer les volumes des petites pyramides centrées sur le squelette et qui s'appuient sur les graines (voir figure 7):

$$V_i = \sum_{s \in S_i} k_i d(s, P_i)^3$$

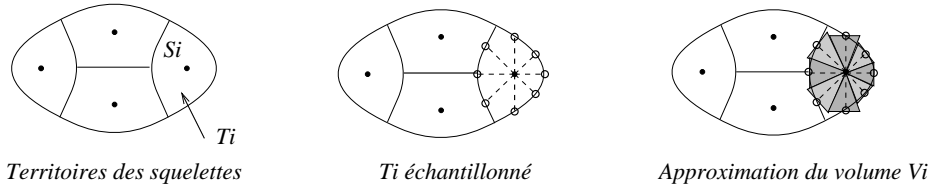


FIG. 7 - Territoires de squelettes et calcul du volume grâce aux graines.

Correction du volume

A partir de cette approximation du volume, on peut gérer les variations de volume de chaque squelette grâce à un contrôleur Proportionnel-Dérivé. On peut en effet changer la fonction d'enrobage de chaque primitive de façon à équilibrer les variations observées lors de l'échantillonnage [DG95]. Pour ne pas modifier les paramètres physiques de notre objet qui sont stockés dans les variations de la fonction potentiel f , on agira sur l'enrobage par une simple translation de f . Ainsi, en cas de baisse de volume, on translatera f vers la droite de façon à augmenter l'épaisseur d'enrobage.

Nous avons choisi de contrôler la dérivé de la translation plutôt que la translation elle-même, afin d'obtenir des variations de volume plus régulières. Notre contrôleur pour chaque squelette s_i prend alors comme entrées à l'instant t :

$$\Delta_i^t = \frac{V_i^t - V_i^0}{V_i^0} \quad \dot{\Delta}_i^t = \frac{V_i^t - V_i^{t-dt}}{V_i^0 dt}$$

et rend la valeur de la dérivé de la translation par rapport au temps à l'instant t , $\dot{\epsilon}_i^t$:

$$\dot{\epsilon}_i^t = \alpha \Delta_i^t + \beta \dot{\Delta}_i^t$$

La valeur du potentiel $f(\|P - s_i\|)$ est donc remplacée par : $f(\|P - s_i\| - \epsilon_i^t)$. Ainsi, on est assuré de contrôler le volume localement comme le montre la figure 8 où deux squelettes ponctuels se rapprochent l'un de l'autre, avec ou sans contrôle de volume. L'écart est considérable : les variations de volume sans contrôleur montent à 400% du volume initial.

Cette méthode est bien évidemment aussi bien utilisable pour le maintien d'un volume constant que pour des variations de volume imposées par l'utilisateur. Il suffit pour cela de faire varier la valeur objective V_i^0 au cours du temps.

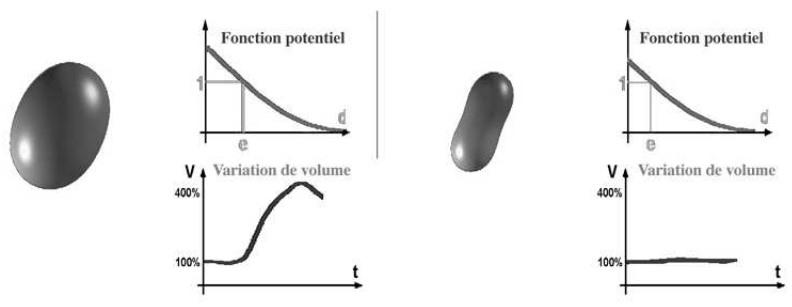


FIG. 8 - Deux squelettes en mouvement se mélangent : à gauche sans contrôleur, à droite avec contrôleur.

5 Résultats

Une grande variété de comportements peut être obtenue en combinant la couche implicite avec différentes structures internes. Nous présentons deux exemples qui illustrent bien cette diversité.

5.1 Animation d'objets munis de squelettes articulés

Les différentes méthodes développées jusque là se prêtent très bien à l'animation de personnages simplifiés, constitués de squelettes articulés recouverts d'une chair déformable. La figure 9 illustre la modélisation fine des contacts, y compris des auto-collisions, pour ce type de modèle. Les articulations de la structure interne ont été simulées grâce à la méthode des contraintes de déplacement [GG94].



FIG. 9 - Un personnage simplifié tombe, entraînant des auto-collisions.

5.2 Substances hautement déformables

Le cas plus complexe des substances pouvant subir des déformations importantes ainsi que des changements de topologie (fractures, fusions) fait intervenir l'ensemble des techniques décrites précédemment : il faut à la fois un échantillonnage performant, un contrôle du volume, et une bonne gestion des mélanges indésirables. La structure interne est formée d'un système particulière, choisi pour sa bonne modélisation des propriétés visco-plastiques [LJF⁺91, MP89].

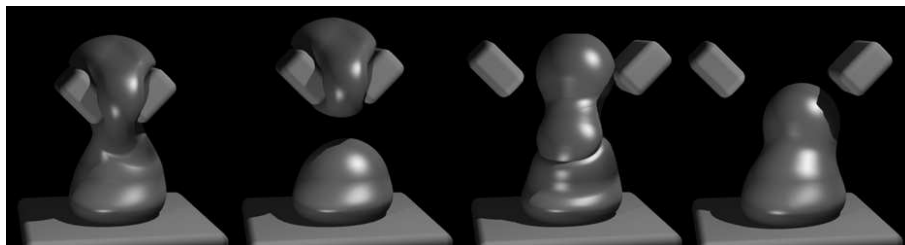


FIG. 10 - Une substance saisie par des pinces, puis relâchée.

La figure 10 montre une substance saisie à l'aide de pinces, puis relâchée, entraînant dans un premier temps une séparation, puis une fusion des deux morceaux obtenus (la possibilité de fusion a

été ajoutée au modèle en redéfinissant la liste de mélange des squelettes en cas de forces de contact dépassant un certain seuil). L'exemple ci-dessous montre deux pâtes fusionnant progressivement sous l'action d'un rouleau rigide.



FIG. 11 - *Fusion progressive par compression*

6 Conclusion

Ce papier a présenté une approche générale qui permet d'enrichir n'importe quel modèle physique existant en l'interfaçant avec un enrobage implicite définissant la surface de l'objet. La représentation implicite utilisée permet une détection et une modélisation fine et efficace des collisions entre objets, ainsi que des auto-collisions entre différentes parties d'un même objet. Les variations de volume peuvent être contrôlées pendant l'animation. Utilisée avec succès pour la simulation de substances hautement déformables, cette approche paraît également prometteuse pour l'animation de personnages. En particulier, le contrôle local de volume pourrait être utilisé pour simuler la contraction des muscles. Dans ce cadre, des essais ont d'ailleurs été faits pour la modélisation et l'animation de lèvres, en traitant les collisions de ces dernières avec les dents ou une cigarette [Tsi95].

Références

- [CHP89] J.E. Chadwick, D.R. Haumann, and R.E. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3):243–252, July 1989.
- [DG95] Mathieu Desbrun and Marie-Paule Gascuel. Animating soft substances with implicit surfaces. *Computer Graphics*, August 1995. Proceedings of SIGGRAPH'95 (Los Angeles, CA).
- [DTG95] Mathieu Desbrun, Nicolas Tsingos, and Marie-Paule Gascuel. Adaptive sampling of implicit surfaces for interactive modeling and animation. In *First International Workshop on Implicit Surfaces*, Grenoble, France, April 1995.
- [Gas93] Marie-Paule Gascuel. An implicit formulation for precise contact modeling between flexible solids. *Computer Graphics*, pages 313–320, August 1993. Proceedings of SIGGRAPH'93 (Anaheim, CA).
- [GG94] Jean-Dominique Gascuel and Marie-Paule Gascuel. Displacement constraints for interactive modeling and animation of articulated structures. *The Visual Computer*, 10(4), March 1994.
- [LC92] M. Lin and J. Canny. Efficient collision detection for animation. In *Third Eurographics Workshop on Animation and Simulation*, Cambridge, England, September 1992.
- [LJF⁺91] A. Luciani, S. Jimenez, J-L. Florens, C. Cadoz, and O. Raoult. Computational physics: a modeler simulator for animated physical objects. In *Eurographics'91*, Vienna, Austria, September 1991.
- [MP89] Gavin Miller and Andrew Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *SIGGRAPH '89 Courses 30 notes*, pages 305–309, August 89.
- [MW88] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. *Computer Graphics*, 22(4):289–298, August 1988. Proceedings of SIGGRAPH'88 (Atlanta, August 1988).
- [OM93] Agata Opalach and Steve Maddock. Implicit surfaces: Appearance, blending and consistency. In *Fourth Eurographics Workshop on Animation and Simulation*, Barcelona, Spain, September 1993.

- [PB88] John Platt and Alan Barr. Constraint methods for flexible models. *Computer Graphics*, 22(4):279–288, August 1988. Proceedings of SIGGRAPH’88 (Atlanta, Georgia).
- [TF88] Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformations: Viscoelasticity, plasticity, fracture. *Computer Graphics*, 22(4):269–278, August 1988. Proceedings of SIGGRAPH’88 (Atlanta, Georgia).
- [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, July 1987).
- [Tsi95] Nicolas Tsingos. Modélisation de lèvres parlantes par des surfaces implicites. *Rapport de DEA d’Informatique*, 1995.
- [TW88] Demetri Terzopoulos and Andrew Witkin. Physically based model with rigid and deformable components. *IEEE Computer Graphics and Applications*, pages 41–51, December 1988.
- [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.
- [WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, August 1986.
- [WW89] Brian Wyvill and Geoff Wyvill. Field functions for implicit surfaces. *The Visual Computer*, 5:75–82, December 1989.