

# Edition de maillage par déformation de cage et coordonnées de Green

Arthur Mensch, Paul Vallet, Michaël Weiss

24 avril 2015

## Résumé

Nous présentons dans ce rapport notre implémentation de l'article [1], qui propose une méthode pour déformer un maillage à partir de la déformation d'une cage de contrôle, en utilisant les coordonnées de Green. Nous revenons sur les bases théoriques de la déformation et présentons les fonctionnalités et détails d'implémentations de notre programme.

## Introduction

Pour être utilisable, une interface de modélisation 3D doit permettre à l'utilisateur de modifier un objet en 3D de manière rapide et intuitive. Si les maillages 3D manipulés sont souvent d'une taille considérable, ils n'ont, s'ils représentent des objets physiques, qu'un nombre limité de degrés de liberté. Il importe donc de pouvoir les modifier avec un petit nombre d'interactions de contrôle. Pour cela, les outils de déformation par cage sont très appropriés : l'idée est d'englober un maillage compliqué par un maillage plus simple, et de répercuter les déformations effectuées la cage sur le maillage cible.

Les outils de déformation par cage reposent sur la description du maillage cible dans un certain 'référentiel' lié à la cage. Nous présentons les principes théoriques de cette déformation, en décrivant plus précisément les *coordonnées de Green* qui constituent un des outils mathématiques utilisables, et décrivons l'implémentation et le fonctionnement d'un démonstrateur.

## 1 Principes théoriques

### 1.1 Déformation par cage

L'objectif est de contrôler la déformation d'un grand nombre de sommets par la déformation d'une structure simple. On décrit pour chaque sommet du maillage cible comme une combinaison linéaire des sommets du maillage de contrôle, ainsi que des normales de ses faces (c'est la contribution de [1]). On écrit donc, pour tout sommet  $\eta$  :

$$\eta = \sum_{\mathbf{v}_i \in \mathbb{V}} \phi_i(\eta) \mathbf{v}_i + \sum_{t_i \in \mathbb{T}} \psi_i(\eta) \mathbf{n}(t_i) \quad (1)$$

en notant  $\mathbb{V}$  et  $\mathbb{T}$  les ensembles des sommets et des faces du maillage de contrôle (la cage). On définit ainsi des ensembles de *fonctions coordonnées*.

Ils existent plusieurs familles de fonctions coordonnées : on peut les prendre toutes nulles sauf quatre  $\phi_i$  correspondant à quatre  $\mathbf{v}_i$  non alignés. Il s'agit alors des coordonnées barycentriques dans le repère barycentrique  $(\mathbf{v}_i)_{1 \leq i \leq 4}$ . Cependant ces coordonnées ne sont pas bonnes au sens où les déformations de la cage induisent des déformations non plausibles physiquement sur le maillage.

**Coordonnées de Green** Les coordonnées utilisées dans l'article sont les coordonnées de Green. Elles reposent sur le fait que la fonction identité soit harmonique (i.e. de Laplacien nul), et sur la troisième identité de Green, corollaire direct du théorème de la divergence, qui stipule, pour  $u = \text{Id}$  harmonique et  $G_\eta$  solution fondamentale de l'équation de Laplace  $\{\Delta G_\eta = \delta_\eta\}$  :

$$\eta = \int_{\partial D} \left( \boldsymbol{\xi} \frac{\partial G_\eta(\boldsymbol{\xi})}{\partial \mathbf{n}(\boldsymbol{\xi})} - G_\eta(\boldsymbol{\xi}) \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{n}(\boldsymbol{\xi})} \right) d\sigma_\xi \quad (2)$$

où  $D$  est le maillage cage englobant le maillage cible. La normale  $\mathbf{n}(\boldsymbol{\xi})$  étant constante sur le maillage cible, on dérive de cette égalité une expression de la forme de l'Eq. 1, avec une expression intégrale des fonctions coordonnées, qui s'écrit analytiquement : on a donc un moyen précis de calculer les  $(\phi_i)_i$  et  $(\psi_i)_i$ .

## 1.2 Déformation

L'avantage de cette méthode est que les fonctions coordonnées fournies sont harmoniques. On va pouvoir les utiliser dans la déformation de la cage, avec une légère correction, pour obtenir une déformation quasi-conforme du maillage cible.

On cherche à répercuter la déformation de la cage sur le maillage cible. Soit  $(v'_i)_i$  et  $(t'_i)_i$  les nouveaux éléments du maillage cage. On écrit alors :

$$\eta' = \sum_{\mathbf{v}'_i \in \mathbb{V}'} \phi_i(\eta) \mathbf{v}'_i + \sum_{t'_i \in \mathbb{T}'} \psi_i(\eta) s(t'_i, t_i) \mathbf{n}(t'_i) \quad (3)$$

où  $s(t'_i, t_i)$  est un terme dépendant explicitement des triangles pré et post-déformation.

**Qualité de la déformation** La définition des  $(\phi_i)_i$  et  $(\psi_i)_i$  puis de la fonction  $s$  s'assure alors des éléments suivant pour la déformation  $\eta \rightarrow \eta'$  :

- Toute similitude appliquée à la cage se traduit par une similitude appliquée au maillage cible.
- La déformation du maillage cible est  $\mathcal{C}^\infty$  et quasi-conforme (le cisaillement est encadré, i.e la modification locale des angles est bornée).

## 2 Démonstration

S'il existe des méthodes pour *encadrer* automatiquement un modèle 3D d'une cage de contrôle, celles-ci ne sont pas adaptées à notre problème. En effet il est difficile de définir a priori la complexité de la cage voulue (i.e. le nombre de triangles utilisés) et la proximité avec laquelle la cage doit suivre la surface extérieure du modèle. Ainsi, pour chaque modèle, la cage de contrôle a été dessinée manuellement, à l'aide du logiciel **Blender**, afin de pouvoir réaliser facilement les déformations voulues.

### 2.1 Fonctionnalités utilisateur

Afin de montrer l'intérêt d'une telle méthode, il était donc nécessaire de pouvoir appliquer des déformations intuitives pour l'utilisateur. Pour cela deux grands problèmes surviennent : la **sélection** des points de la cage que l'on cherche à déformer et la **déformation** à proprement parler. Nous sommes donc partis du moteur graphique utilisé à partir du deuxième TP où les mouvements de caméra contrôlables grâce à la souris étaient déjà implémentés.

**Sélection** La sélection peut se faire de deux manières différentes : une sélection triangle par triangle et une sélection par carré.

La sélection triangle par triangle s'effectue en pointant un triangle à l'aide de la souris, un rayon est alors tracé entre le *near plane* et le *far plane* ayant pour coordonnées dans le plan celles de notre souris. Ce rayon nous permet alors de déterminer le triangle ayant une intersection avec ce rayon et étant le plus proche du *near plane*. Celui-ci est alors ajouté à l'ensemble des triangles sélectionnés. Inversement, si ce triangle visé était déjà sélectionné, il est alors enlevé de l'ensemble des triangles sélectionnés.

La sélection par carré, elle, permet de dessiner un carré sur l'écran de l'utilisateur. Alors, tous les triangles projetés dans le *near plane* étant contenus dans ce carré seront ajoutés à la sélection, ce qui permet de sélectionner facilement toute une zone de notre cage de contrôle.

Une fois ces triangles sélectionnés, il est alors possible de les modifier, ce qui résulte en une modification du modèle 3D.

L'interface est faite pour être la plus intuitive possible et repose sur le même modèle d'interaction que **Blender** : on navigue rapidement entre les différents modes par des touches, un clic gauche valide la transformation et un clic droit l'annule. Les différents modes de transformation sont identifiés par une couleur variable de la sélection.

**Translation** L'ensemble des triangles sélectionnés de notre cage peut être translaté en passant en mode translation. Cette translation s'effectue dans le plan parallèle au plan de l'écran (*near plane*) et suit les mouvements de la souris de l'utilisateur jusqu'à ce que celui-ci mette fin au mode de translation.

Pour pouvoir permettre d'effectuer des modifications plus précises, il est aussi possible d'effectuer une translation point par point en sélectionnant un point parmi ceux des triangles sélectionnés.

**Homothétie** Il est aussi possible pour l'utilisateur d'effectuer une homothétie calculée à partir du barycentre des points sélectionnés, dont le ratio est donné entre la différence de position de la souris par rapport à la projection du barycentre à l'écran entre deux instants.

**Rotation** Toujours en prenant comme référence l'évolution de la position de la souris par rapport à la projection du barycentre, il est possible d'effectuer une rotation autour de ce barycentre dans le plan parallèle au plan de l'écran (*near plane*).

## 2.2 Fonctionnement algorithmique

Les déformations sont basés sur la théorie présentée en section 1. Les algorithmes de mise à jour des coordonnées et de gestion de la sélection sont encapsulés dans une classe `BoundingBoxMesh`.

**Calcul des coordonnées** Au chargement du maillage, on calcule les coordonnées de Green de chacun de ses sommets, puis on remplace les sommets donnés en entrée par les sommets calculés à partir de ces coordonnées. On utilise dans cette phase les algorithmes présentés en [1].

**Mise à jour** Une fonction de mise à jour totale calcule la position de chaque sommet en effectuant le calcul entier de l'Eq. 3. Cependant, lorsqu'on agit seulement sur une partie de la cage, on peut effectuer une mise à jour des position en ligne, en ne modifiant qu'une partie de la somme de l'Eq. 3.

Pour permettre une mise à jour en temps réel, on a donc recours à la stratégie suivante :

- Lors de la sélection, on sauvegarde les positions des sommets actuelles en leur soustrayant la partie de la somme concernée par la partie de la cage sélectionnée.
- Lors du déplacement, on met à jour le mesh actuel à partir de cette sauvegarde, auquel on ajoute la partie de la somme concernée par la sélection, dans la nouvelle configuration.
- Lorsqu'on cesse le déplacement, les positions sont recalculés par mise à jour totale (pour éviter les dérives numériques).

**Optimisation** Les calculs de position pour chacun des sommets peuvent être parallélisés. Par souci de simplicité, nous avons simplement utilisé les annotations de `OpenMP`, ce qui permet un gain de performance de l'ordre d'un facteur 5 sur un processeur à quatre coeur. On peut envisager de paralléliser ce calcul sur GPU. Nous avons essayé d'optimiser les calculs en utilisant les instructions `SSE` pour les opérations sur les vecteurs, mais la migration s'est montrée plus compliquée que prévu.

## 2.3 Résultats

Notre démonstrateur permet d'éditer la cage de manière relativement complète et intuitive, à la manière de la fonction d'édition de maillage du logiciel `Blender`. La méthode utilisée permet, avec un code optimisé, de montrer à l'utilisateur la déformation du maillage cible en quasi *temps réel*, ce qui est particulièrement satisfaisant en terme d'ergonomie.

## Conclusion

Notre démonstrateur expose la puissance des coordonnées de Green pour déformer des maillages de manière intuitive et physiquement plausible. Cette méthode peut notamment être utilisée pour animer des maillages en un temps raisonnable. Un certain nombre d'optimisations peuvent encore être effectuées en portant les calculs sur GPU. Il serait intéressant d'intégrer cet outil comme *plug-in* de `Blender`, qui jouit déjà d'une interface utilisateur intuitive et documentée.

## Référence

Le démonstrateur est téléchargeable à l'adresse <https://github.com/arthurmensch/ProjectINFSI350>.

## Références

- [1] Yaron Lipman, David Levin, and Daniel Cohen-Or. Green coordinates. In *ACM Transactions on Graphics (TOG)*, volume 27, page 78. ACM, 2008.