TP Déformation et animation de maillages *CPE*

durée - 6h

Décembre 2009

Ce travail se scinde en deux parties.

Partie 1 : *Introduction à la déformation de maillage* (ie. expression analytique d'une déformation exprimée pour chaque sommet).

Objectifs :
- Se familiariser avec la structure de donnée classique d'un maillage (nécessaire à son affichage efficace par OpenGL/DirectX).
- Manipuler les structures algorithmiques associées.
- Appliquer des déformations animées.
- Aborder les problématiques de collisions, lissages.

Requis : 1.1 et 1.2. (1.3 est en extra)

Partie 2 : *Introduction à l'animation par squelette (skinning lisse).*

Objectifs :
- Comprendre l'application des déformations rigides (notamment rotations).
- Mettre en place une déformation locale.
- Débutter la mise en place d'une déformation lisse.

Requis : 2.1, 2.2 et débutter 2.3, voir 2.4.

1 Manipulation de maillages, déformation de surfaces.

1.1 Préliminaires

- Executez le programme et vérifiez que vous comprenez les différentes commandes.
- Prêtez attention aux fonctions *load model*, *evolve*, *draw extra* du fichier main.

1. *load model* charge un fichier off en mémoire et prépare les structures de données globales avant la boucle d'affichage. La fonction n'est exécutée qu'une unique fois au début.
2. *evolve* permet de modifier les structures au cours du temps. Elle contient un timer t evoluant entre 0 et 1 automatiquement. La fonction est automatiquement appelée à intervals fixes.
3. *draw extra* permet d'afficher des éléments supplémentaires dans la scène. On utilisera par la suite la fonction *draw sphere (centre,rayon)* dans celle-ci.

code [1] : Quel maillage chargez vous ? Le nombre de sommet à traiter est-il correct vis à vis de votre matériel. Chargez d'autres maillages et observez l'influence sur les performances.

code [2] : Observez la classe *mesh* et completez la methode *get vertex*, *set vertex*, *get connectivity* et *set connectivity* pour aider à la manipulation des données.

code [3] : En utilisant la fonction *evolve*, déplacez l'un des sommets de votre maillage en fonction du temps.
Déplacez plusieurs sommets à l'aide de fonctions périodique classiques.

1.2 Déformation normale

Dans cette partie, on s'intéresse à la déformation de sommets le long de la normale à la surface. À partir d'une surface X de coordonnées $\mathbf{x}(u, v)$, le but est de construire la surface Y, de coordonnées

$$\mathbf{y}(u, v) = \mathbf{x}(u, v) + \lambda(t)\,\mathbf{n}(u, v) ,$$

où $\mathbf{n}$ est la normale unitaire à X, et λ un paramètre évoluant au cours du temps.

- Exprimez $\mathbf{y}$ analytiquement en fonction de $\mathbf{x}$,u et v.
- Peut-on qualifier cette déformation de linéaire ?
- Rappellez la méthode d'approximation de calcul des normales dans le cas discret. Afin de calculer ces normales en chaque sommet, il faut construire une structure de données permettant de stocker l'indice des faces adjacentes à un sommet donné (1-star).

  Dans les cas de requêtes suivantes, quelle structure de données vous parrait la plus appropriée. Estimez les complexitées associées :
  1. Connaitre le nombre de faces adjacentes à un sommet i.
  2. Savoir si le sommet i est adjacent à la face j.
  3. Savoir si le sommet i et j sont adjacents à une même face.
  4. Acceder au *1-star* d'un maillage obtenu comme le polygone convexe d'une distribution de 2 millions de sommets repartis suivant une loie Gaussienne.
  5. Mettre à jour votre structure en ajoutant au sommet i une adjacence avec la face j.
  6. Supprimer trois sommets consecutifs du maillage et mettre à jour la structure.

code [4] : Implémentez cette structure de voisinage sous la forme d'un
`vector<set<int> >`

code [5] : Utilisez cette structure pour calculer les normales en chaque sommet et appliquez la déformation demandée

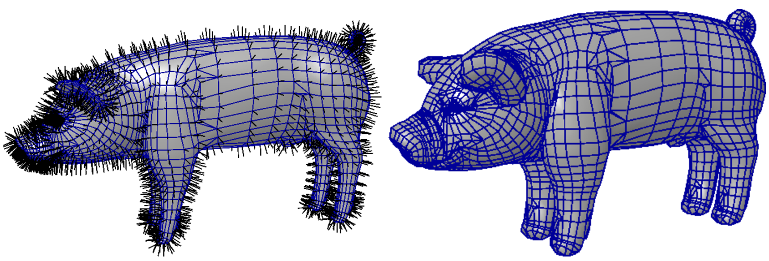

FIG. 1 – Maillage avant-après déformation

1.2.1 Analyse

- Commentez quantitativement l'évolution de la courbure (Gauss) de la surface en fonction de λ.
- Quel est l'image des surfaces suivante par cette transformation de paramètre λ :

  1. Sphere de rayon r.
  2. Tore de rayons r et R.
  3. Cube de coté a (attention plusieurs cas possibles).

- D'après ces résultats, pensez vous pouvoir utiliser ce type de déformation pour effectuer un lissage de surface.

1.3 Lissage Laplacien

Uniquement si vous êtes très en avance et que vous vous ennuyez.

1.3.1 Cas continu

Le lissage Laplacien consiste a approximer l'equation de diffusion sur les coordonnées d'une surface $\frac{\partial f}{\partial t} = \lambda \triangle f$.

1.3.2 Cas discret

Soit $\mathcal{V}_i$ le 1-voisinage du sommet i.

L'opérateur $\triangle$ est approximée en discret par $\frac{1}{\text{card}(\mathcal{V}_i)} \sum_{j \in \mathcal{V}_i} (\mathbf{x}_i - \mathbf{x}_j)$. L'equation de diffusion discrète s'écrit donc

$$\forall i \ , \ \mathbf{y}_i = \mathbf{x}_i + \frac{\lambda}{\text{card}(\mathcal{V}_i)} \sum_{j \in \mathcal{V}_i} (\mathbf{x}_i - \mathbf{x}_j) \ .$$

- Peut-on qualifier cette deformation de linéaire ?
- Réecrivez l'équation en terme de combinaison convexe entre la position courante et le barycentre du 1-voisinage.

1.3.3 Application

code : Implémentez cette déformation, faites dépendre λ du temps. Quels est son domaine de variation ? Appliquez la déformation iterativement N fois. Que ce passe-il si λ dépasse sont domaine d'application ? est-ce prévisible ?

2 Animation - Introduction à l'animation par squelette.

2.1 Prise en main

– La classe *MC matrix* permet de créer une matrice de rotation (voir cours) définie par un axe **u** avec un angle θ par l'appel

```
//exemple rotation de pi/2 suivant l'axe (x,y,z)=(1,0,0)
MC_v3d u(1,0,0);
double theta=M_PI/2.0;

//matrice
MC_matrix R=MC_matrix::rotation(u,theta);

//application de la rotation sur x0
MC_v3d x0(5,3,2);
MC_v3d x=R*x0;
```

Cette rotation sera centrée à l'origine. Si le centre de rotation est donné par un autre point **c**, la syntaxe est la suivante :

```
MC_v3d c(5,4,-5.5); //centre de rotation
MC_v3d x=MC_v3d(R*(x0-c))+c;
```

code [6] : Appliquer une rotation globale sur tout vos sommets. Faites varier l'angle, puis l'axe, puis le centre de rotation en fonction du temps.

2.2 Animation locale

code [7] : Faites désormais varier l'angle de la rotation en fonction des coordonnees du maillage.

code [8] : Imiter l'animation de deformation de la partie avant du cochon vue en cours (vous devrez rendre votre deformation locale).

code [9] : Selectionner les sommets situées a l'interieur d'une sphere que vous definirez pour ne considerer que les sommet de la tête d'un animal de votre choix. Animez cette partie uniquement.
rem. On pourra s'aider de l'appel draw sphere

```
ex.

void draw_extra()
{
  draw_sphere(MC_v3d(1,1,0),0.5);
}
```

2.3 Déformation lisse

– La deformation est rigide par morceaux. Les jonctions sont donc soumises aux auto-collisions et élongations importantes.

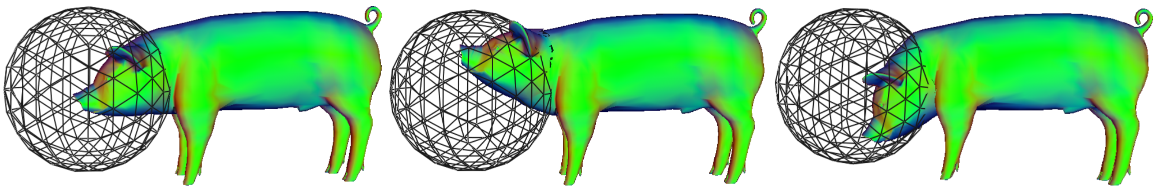

FIG. 2 – Cas rigide : Sphere d'influence avec pose-original, pose extreme1, pose extreme 2

Afin de lisser la deformation, définissez un champ scalaire $f(\mathbf{x})$ continu à l'interieur de votre sphere d'influence (centre $\mathbf{c}$ et rayon r) tel que :

- $f = 1$ au centre de la sphere ($\mathbf{x} = \mathbf{c}$)
- $f = 0$ sur la sphère ($\|\mathbf{x} - \mathbf{c}\| = r$).

On pourra penser à utiliser

$$f(\mathbf{x}) = 1 - \left(\frac{\|\mathbf{x} - \mathbf{c}\|}{r}\right)^{\beta} .$$

code [10] : Implémentez un champ de matrices variant continument entre votre rotation principale R et l'identitée Id en utilisant une interpolation linéaire de matrices à l'aide de f.

Votre deformation sera alors du type

$$\mathbf{y} = \Big(f(\mathbf{x})\mathrm{R} + [1 - f(\mathbf{x})]\mathrm{Id}\Big)(\mathbf{x} - \mathbf{x}_0) + \mathbf{x}_0 ,$$

où $\mathbf{x}_0$ est le centre de rotation associé à R.

– Les matrices de déformation sont-elle toujours des matrices de rotations ? Commentez.

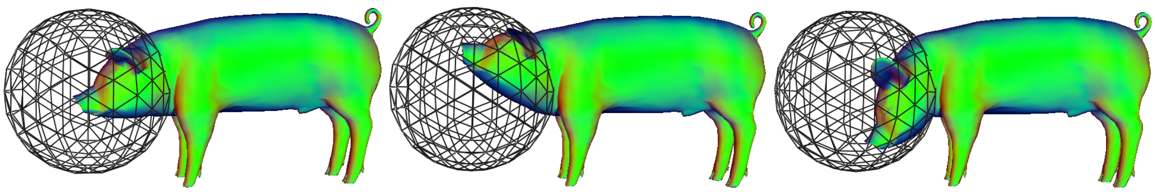

FIG. 3 – Déformation lisse : Sphere d'influence avec pose-original, pose extreme1, pose extreme 2

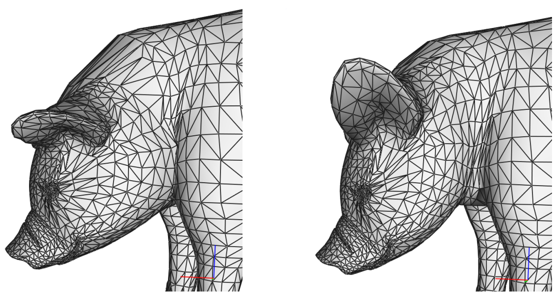

FIG. 4 – Comparaison entre cas rigide et cas lisse sur la déformation du maillage.

2.4 Hierarchie

Si vous avez fini :

Le but est de d'animer un mouvement de tête d'un animal en même temps que son corps.

code [11] : Tout en gardant l'animation de la tête de l'animal, animez une sous partie de sa tête à l'aide de la même méthode (ex. son museau).

On remarquera que dans ce cas, la deformation se met sous la forme

$$\mathbf{y} = \mathrm{R}_1\Big(\mathrm{R}_2(\mathbf{x} - \mathbf{c}_2) + \mathbf{c}_2 - \mathbf{c}_1\Big) + \mathbf{c}_1 \; .$$

rem. on pourrait l'exprimer plus simplement si on utilise des matrices 4×4*.*

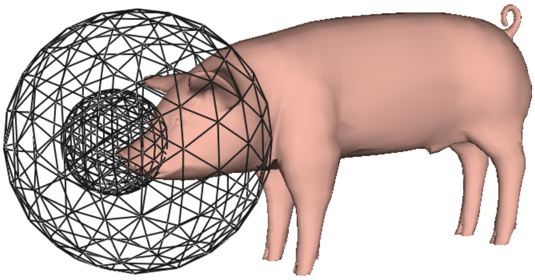

FIG. 5 – Spheres d'influences du mouvement complet.

- Le skinning de personnage est la généralisation de cette méthode pour un squelette d'animation (donnant les centres des matrices de rotations) défini comme une suite hérarchique de positions. Deux idées de poursuite :
  1. Construire un véritable squelette d'animation hierarchique.
  2. Automatiser le calcul des facteurs d'influences (poids de skinning) par calcul de distances normalisées au squelette.