Projets de programmation

Olivier Devillers François Sillion

à choisir avant le 5 Novembre 2003

Quatre séances d'accompagnement du projet sont proposées, les 22 octobre, 12 Novembre, 3 Décembre et 10 Décembre. La première séance (le 22 octobre) vous permettra de demander des explications sur les sujets, voire de démarrer votre travail.

Pour la soutenance (semaine du 15 décembre) vous aurez à fournir un rapport bref (4 pages) et une page HTML, et à faire un exposé de 20 minutes environ intégrant une démo.

Merci de confirmer dès que vous avez choisi votre sujet, par e-mail (`Olivier.Devillers@inria.fr`,`Francois.Sillion@imag.fr`), aux 2 enseignants quel que soit le sujet, SVP.

Vous êtes invités à vous concerter au préalable pour éviter les accumulations sur quelques sujets. Si plusieurs élèves choisissent un même sujet, il leur sera demandé de traiter des applications différentes ou d'approfondir chacun un point particulier.

1 Simplification de terrain

Étant donné un ensemble de points représentant un terrain, on utilise souvent la triangulation de Delaunay de la projection horizontale de ces points pour relier ces points et obtenir un modèle polyédrique du terrain.

En utilisant la suppression d'un point dans la triangulation de Delaunay on simplifiera le terrain en supprimant des points tant que la distance de Hausdorff "verticale" entre le terrain initial et le terrain simplifié reste inférieure à un certain seuil. (La distance de Hausdorff verticale entre deux terrains est le maximum de la distance verticale entre les deux terrains, on pourra se contenter de prendre le maximum des distances verticales en tous les sommets des deux surfaces plutôt que sur tous les points).

Deux fichiers de données sont disponibles

ftp ://ftp-sop.inria.fr/prisme/devillers/dem.txt.gz

ftp ://ftp-sop.inria.fr/prisme/devillers/rivers.txt.gz

2 Delaunay par division fusion

Programmer l'algorithme de division fusion vu en cours en mettant en evidence les différentes phases de son déroulement sur l'affichage graphique (de toute façon un tel affichage est utile pour debugger).

3 Crust

Le crust est une méthode pour reconstruire une approximation du contour d'un objet 2d ou de la surface d'un objet 3d à partir d'un ensemble de points mesurés sur un objet. En dimension 2, la méthode consiste à :
- construire la triangulation de Delaunay et les sommets du diagramme de Voronoï de l'ensemble des points mesurés,
- construire la triangulation de Delaunay de l'ensemble des points mesurés augmenté de l'ensemble des sommets de Voronoï précédemment calculés,
- ne conserver de cette dernière triangulation que les arêtes joignant deux points mesurés.

Ecrire un programme pour calculer le crust. Essayer la méthode sur différents ensembles de points relevés sur des contours simples.

4 Triangulation de Delaunay et planification de trajectoire

Étant donné un ensemble d'obstacles ponctuels, le diagramme de Voronoï permet de répondre à la question «quel est le plus proche obstacle ?»

Si on a maintenant un robot circulaire de rayon 1, il est clair qu'une position pour le centre du robot sera valide si le plus proche obstacle (et donc tous les autres aussi) est à une distance plus grande que 1.

Étant donné un ensemble d'obstacles ponctuels on écrira un programme qui calcule (et affiche)

— le bord de l'ensemble des positions admissibles pour le robot

— un chemin entre une position de départ et d'arrivée.

5 Paires antipodales

Une droite est dite support d'un ensemble convexe de E^2 si :

1) son intersection avec le convexe est non vide

2)le convexe est contenu dans l'un des deux demi-espaces (fermés) définis par cette droite.

Deux sommets d'un polygone convexe sont dits antipodaux si ils appartiennent à deux droites supports parallèles.
- Etant donné un polygone convexe, construire et visualiser l'ensemble des paires antipodales.

- **Application : diamètre d'un ensemble de points.** On appelle diamètre d'un ensemble de points la distance maximale entre deux points de l'ensemble. Le diamètre d'un ensemble de points est réalisé par une paire de sommets antipodaux de l'enveloppe convexe. Calculer et visualiser le diamètre d'un ensemble de points.
- **Application : largeur d'un ensemble de points.** On appelle largeur d'un ensemble de points la plus petite distance entre deux droites parallèlles limitant une bande qui contient tous les points. La largeur d'un ensemble de points est réalisée par deux droites supports parallèlles de l'enveloppe convexe dont l'une contient une arête de l'enveloppe convexe. Calculer et visualiser la largeur d'un ensemble de points.

6 Simplification de maillages

On se propose de réaliser un système de simplification de maillages, pouvant être utilisé sur des objets (polyèdriques) 3D ou sur des maillages 2D / 2D-1/2 représentant par exemple des images ou des terrains.

On suppose donc toujours que l'on dispose d'un maillage dense, c'est-à dire d'un ensemble de triangles formés à partir d'un ensemble de points. On cherchera à définir un opérateur de simplification fonctionnant par élimination d'une arête du maillage, ainsi qu'un ou des critères de mise en oeuvre *progressive* de cet opérateur. En d'autres termes, on se pose la question du choix de l'arête à éliminer. On définira également l'opérateur dual d'enrichissement, qui dédouble un sommet pour re-créer une arête.

On pourra alors définir un modèle (3D ou 2D) comme une séquence d'opérations d'enrichissement associées au modèle de base. Cette représentation permet alors de sélectionner un niveau de détail en spécifiant le nombre de triangles à obtenir.

7 Visualisation de modèles implicites

Par opposition à une représentation *explicite* des formes, sous forme de polyèdres ou de surfaces paramétriques, les modèles *implicites* sont donnés comme une fonction mathématique $F(x, y, z)$ définie dans l'espace 3D, et dont on extrait une iso-surface (en général on s'arrange pour que la surface recherchée soit le lieu de $F(x, y, z) = 0$).

La visualisation de ces surfaces pose des problèmes particuliers : dans ce projet on réalisera une triangulation d'une surface implicite à partir d'un échantillonnage régulier de l'espace, qui permet d'identifier un certain nombre de points de passage de la surface.

On pourra aussi s'intéresser à la visualisation par tracé de rayons, en recherchant des algorithmes fiables d'intersection d'une surface implicite avec un rayon.

8 Interpolation de forme 3D

Le problème du passage "continu" d'une forme 3D à une autre est important pour des applications comme les effets spéciaux du cinéma (cf. Terminator 2) ou le lissage entre différents niveaux de détail dans la représentation d'un objet.

On pourra commencer par étudier le passage d'un polyèdre à un autre. Dans le cas de topologies simples (i.e sphériques) un algorithme simple consiste à projeter l'ensemble des sommets des deux polyèdres sur une sphère, et à réaliser une triangulation de l'union de ces ensembles de points. Cette triangulation sert de base à la déformation progressive d'une forme à l'autre.

On posera le problème de la mise en correspondance des deux polyèdres pour obtenir une transformation plus réaliste (par exemple si les deux formes sont deux versions orientées différemment du même objet : on ne souhaite pas faire un morphing "brutal", mais effectuer une rotation de l'objet).

9 Rendu à base d'images

Dans ce projet, on s'intéresse à la création d'images à partir d'images (de synthèse). On suppose donc que l'on dispose d'images ainsi que des données de profondeur associées dans un tampon de profondeur (pour réaliser les mêmes opérations avec des images réelles, il y aurait la difficulté supplémentaire d'obtenir cette information de profondeur). On étudiera de quelle façon on peut interpoler la vue entre deux images, par reprojection des pixels des images de référence pour synthétiser une nouvelle vue. Attention : pour ce projet il faudra écrire le programme (certes simple) de visualisation, à l'aide de la bibliothèque OpenGL.

10 Simulation de l'éclairage par la méthode de radiosité

Ce projet vise à mettre en oeuvre une méthode de calcul de l'éclairage (appelée méthode de radiosité) qui modélise l'ensemble des interactions lumineuses, y compris les réflexions multiples entre surfaces. On se donne une scène 3D composée d'un certain nombre de facettes planes. Chaque facette est supposée diffuse, munie d'une réflectance et d'un flux émis (non nul pour les sources lumineuses).

On exprime l'équilibre des échanges d'énergie radiative entre ces surfaces par un système d'équations linéaires, dont les coefficients font intervenir des termes géométriques, les "facteurs de forme".

La simulation implique donc de calculer les facteurs de forme, dans un premier temps, puis de résoudre le système linéaire. On obtient alors une "couleur" pour chaque face et l'on peut effectuer la visualisation.