Examen de Visualisation-Multiresolution
Option Multimédia - *Polytech*

17 decembre 2008
Durée 2h

1 Consignes

- Polycopiés, notes de cours, listing de TP autorisés.
- Calculatrices graphiques autorisés.
- Livres non autorisés.
- Communication avec l'exterieur interdite.
- Ordinateurs interdits.

Pour rapporter la totalité des points, toutes vos réponses devront être jutifiés clairement à l'aide de phrases construites en français ou anglais.

L'utilisation de schémas explicatifs sera grandement apprécié.

En cas de doute sur le sens des questions, explicitez clairement les suppositions que vous faites. Si vous décelez ce qui vous semble être une erreur d'énoncé perturbant vos réponses, vous l'expliquerez dans votre copie et poursuivrez par les autres questions.

Pour certaines questions, plusieurs réponses peuvent être acceptés à divers niveaux de pertinences. Dans ce cas, vous pouvez réaliser un rapide comparatif en indiquant les avantages et inconvénients des différentes approches que vous proposez.

Vous êtes libres de créer vos propres variables intérmédiaires afin de répondre le plus précisément possibles à certaines questions.

L'examen comporte 7 pages.

Les questions auxquelles vous devez répondre sont numérotés de 1 à 19 et différenciés du reste du texte à l'aide d'un point noir •

Vous pouvez répondre aux questions dans l'ordre souhaité de façon indépendante.

Le barème est donné à titre indicatif.

Dans l'ensemble de cet examen, nous nous placerons dans l'espace $\mathbb{R}^3$ décrit par les coordonnées cartésiennes classiques $\overrightarrow{x} = (x, y, z)$.

Un vecteur $\overrightarrow{v}$ de l'espace $\mathbb{R}^3$ seront dénotés par cette notation classique et ces trois composantes seront donnés par (v_x, v_y, v_z). La norme Euclidienne sera dénoté par $\|\overrightarrow{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$.

2 Modélisation Polygonale (1.5 points)

Nous souhaitons modéliser un paysage montagneux.

Nous disposons pour cela d'un ensemble de $N_x \times N_y$ points répartis uniformément. Chaque point i possède des coordonnées (x_i, y_i, z_i).

Les coordonnées x_i et y_i varient entre $[0, 1]$, et les valeurs z_i vous sont données dans un fichier externe.

- 1. Les coordonnées z définissent donc un champ particulier. Quel est le nom formel de ce type de ce champ ? De combien de variables dépend-il ?

Nous souhaitons dans un premier temps observer l'aspect général des montages sous forme polygonale grossière comme montré en fig. 1.

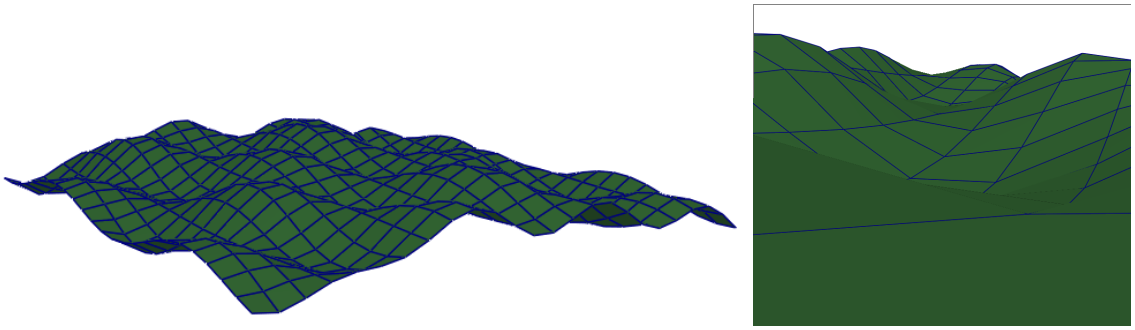

FIG. 1 – Exemple d'un paysage polygonale

- 2. Proposez une méthode permettant de visualiser ces montagnes rapidement (visualisation OpenGL ou export en un format de fichier polygonal).
  Ecrivez notamment un pseudo code mettant en jeu les structures de données utilisés afin de construire ces polygones et/ou illustrez votre réponse.

3 Modélisation Surfacique (2.5 points)

Nous souhaitons maintenant obtenir une surface lisse sous forme d'une approche multiresolution.

- 3. Au vue des données expliquez en quoi l'approche B-Spline vous semble intéressante. Que pensez vous de l'utilisation des patchs de Béziers dans ce cas ?
  Dans le cas où vous envisageriez par la suite d'utiliser une édition hierarchique sur les sommets de votre terrain, garderiez vous cette même méthode ? Le cas échéant, par quelle autre méthode la remplaceriez vous ? Justifiez.

En dehors du terrain montagneux, vous souhaitez placer un océan englobant dont la hauteur z est fixée à $z = 0$.

- 4. Afin d'éviter toute discontinuité entre la montagne et l'océan au niveau des bords, à quelle hauteur devez vous fixer les sommets du bords et comment les disposez vous ?

- 5. En considérant que chaque patch B-Spline sera subdivisé en $M_x \times M_y$ sommets, combien de sommets seront finalement visualisés pour la scène entière ?

4 Multiresolution (1.5 points)

On supose que l'on dispose à tout moment d'une caméra se déplacant sur ce terrain en suivant son altitude. Le terrain étant grand, il vous semble inutile d'afficher beaucoup de détails dans les régions éloignés.

- 6. En fonction de la position de la caméra dans votre grille de départ, proposez une méthode subdivision plus approprié prenant avantage de la modélisation par surfaces B-Spline.

5 Coloration (2 points)

Vous souhaitez désormais affecter une coloration à ce terrain en définissant une composante rouge, verte et bleue pour chaque sommet affiché. Vous supposez que les régions devront satisfaire les contraintes suivantes :

* $z = 0$: couleur verte symbolisant de la végétation.
* $z = Z_1$: couleur brune symbolisant des roches.
* $z = Z_2$: couleur gris-blanc symbolisant des monts enneigés.

- 7. Vous souhaitez obtenir un dégradé de couleurs lisses. Écrivez l'équation que vous utiliseriez pour définir la couleur (R, V, B) d'un sommet de coordonnées (x_i, y_i, z_i).

Vous vous rendez finalement compte que les données de couleurs attendues vous ont été fourni dans un fichier séparé pour chaque point de la grille initiale.

- 8. Comment appliquez vous ces couleurs sur vos sommets construits par la méthode B-Spline ?

- 9. D'après vous, quelle méthode de coloration donnera le résultat le plus réaliste ?

6 Textures et subdivision (4.5 points)

Afin d'éviter les grands aplats de couleurs sur votre terrain, vous suggérez de lui donner un aspect plus détaillé en appliquant une texture monochromatique qui fera varier la luminance des couleurs comme montré en fig. 3.

FIG. 2 – De gauche à droite : Couleur seule, texture seule, couleur + texture mélangé

Vous souhaitez appliquer cette texture en la répétant 4 fois suivant x et y sur chaque patch. On rappel que l'adressage d'un texel dans une image est normalisé entre $[0, 1]$.

- 10. Quels sont les coordonnées de textures associés à un sommet i ? On pourra notamment appeler (k_{xi}, k_{yi}) l'indice de ce sommet dans le patch courant. Quelle méthode mettez vous en place pour éviter les discontinuités ?

On souhaite désormais déplacer un personnage à l'écran. Pour cela, la tête du personnage sera modélisé par un simple cube.

On plaque sur ce cube le visage d'un personnage sur au moins cinq faces afin de donner une impression de $3D$. On peut notamment prendre exemple de la fig. 3 ainsi que du listing section 9 de construction de ce cube en language $c++$.

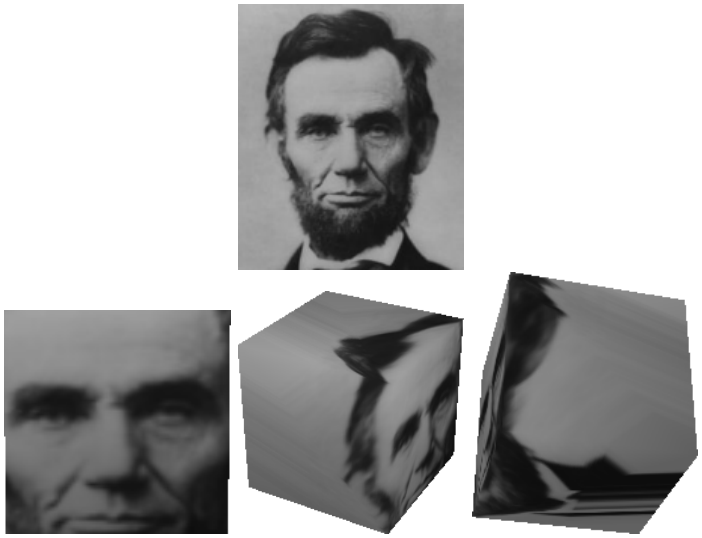

FIG. 3 – Haut : Texture du visage. Bas : Vue du cube texturé sous différents angles.

- 11. Donnez les coordonnées de textures associés à chaque sommet du cube (valeurs (Tx_i, Ty_i) du listing). Illustrez la correspondance du placage du cube sur l'image en deux dimensions. (ici on ne considèrera pas forcément une texture répétitive).

Vous souhaitez désormais subdiviser votre cube défini par 6 faces carrés afin que celui-ci converge vers une forme cylindrique de rayon R et de hauteur h. (les faces hautes et basses restent planes).

- 12. Expliquez votre méthode de subdivision ainsi que les nouvelles coordonnées associées aux points ajoutés.
  Illustrez la subdivision que vous utiliseriez.
  Quelles sont les coordonnées de textures des nouveaux points ainsi introduits ?
  Comment différenciez vous les points à projeter sur les cotés du cylindre de ceux qui ne le sont pas (partie haute et basse) ?
  Combien de sommets obtenez vous à la k^{ieme} subdivision ?

7 Subdivision (5.5 points)

Finalement, un artiste vous fournis une géométrie d'un visage plus complet tel que montré en fig. 4. Les coordonnées de textures sont alors définies manuellement. Vous souhaitez raffiner ce maillage tout en convergeant vers une surface C^2.

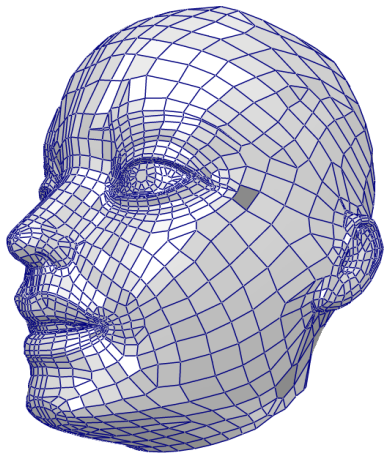

FIG. 4 – Exemple d'une géométrie polygonale d'une figure.

On vous suggère d'utiliser une surface B-Spline sur cet ensemble de points.

- 13. Qu'en pensez vous ?
  Expliquer votre choix, et proposez le cas échéant une autre méthode de type subdivision de faces.

Vous remarquez que ce maillage possède différents types de polygones. Les polygones pouvant aller de 3 jusqu'à N sommets (N pouvant être grand).

- 14. Proposez une structure de données permettant de gérer des polygones ayant un nombre quelconque de sommets sans nécessiter une triangulation préalable.
  Ecrivez une fonction permettant d'acceder à la composante y du sommet s d'un polygone p.

Cette surface vous semblant relativement grossière, vous la subdiviser à l'aide de l'algorithme de Doo-Sabin (corner-cutting).

- 15. Sachant que la surface actuelle est une variété ne possèdant aucun bord ni trous, contient N_f faces et N_s sommets, combien de faces sont obtenue après k subdivisions ?

Nous souhaitons désormais ajouter une "montagne Russe" provenant d'une structure réelle dans le paysage. Pour cela, on définie une surface tubulaire triangulé symbolisant la trajectoire des wagons (on ne s'occupe pas des echaffaudages).

Les sommets de la surface en tube ainsi définie sont positionnés en accord avec une étude de résistance au vent et ne doivent donc pas être déplacés.

On vous fournis en entré un fichier d'un maillage tridimensionnel non régulier quelconque déjà préconstruit. Ce maillage est relativement peu détaillé et vous souhaitez le raffiner afin d'améliorer la visualisation tout en restant cohérent vis à vis des informations structurelles.

- 16. Quelle méthode vous semble judicieuse à utiliser ? justifiez.

8 Visualisation (2.5 points)

Afin d'étudier si votre structure est viable, le bureau d'étude à fait préalablement construire une maquette envoyé en soufflerie. Les données traitées vous sont données sous la forme d'un ensemble de vitesses découlement d'air $\overrightarrow{v} = (v_x, v_y, v_z)$ pris en un instant t. Chaque vitesse correspond à une mesure prise sur une grille rectangulaire régulière de N_x, N_y, N_z positions.

- 17. Vous possédez donc un champ des vitesses autour de votre structure. À quel nom formel ce type de champ cela correspond-il ?

Vous souhaitez avoir un apercu rapide des régions pour lesquelles la vitesse du vent est la plus intense pour identifier les points de fragilités de la structure.

- 18. Quelle méthode proposez vous pour avoir cet apercu rapide ? Expliquez précisément votre démarche ainsi que les variables étudiés permettant d'obtenir une visualisation globale intéressante qualitativement.

Vous souhaitez maintenant plus finement quantifier la limite surfacique de ces régions à haute vitesses définie par $\|v\| \geq v_0$. Vous souhaitez notamment construire une surface triangulée des régions limites.

Aillant entendu parler d'un problème de dépot de brevet vis à vis de l'algorithme du marching cube, vous décidez d'éviter tout risque en construisant votre propre algorithme. Vous décidez donc de ne pas réaliser un découpage en cubes classique, mais un découpage de votre volume en tétraèdres. Vous appelez alors votre algorithme "marching-tetrahedra".

- 19. Expliquez précisément son fonctionnement en vous aidant de schémas. On détaillera notamment les cas possibles rencontrés.

9 Listing

```
std::vector <double> vertex;
std::vector <double> normal;
std::vector <double> color;
std::vector <double> texture;
std::vector <int> connectivity;


vertex.push_back(0);  vertex.push_back(0);  vertex.push_back(0);
vertex.push_back(1);  vertex.push_back(0);  vertex.push_back(0);
vertex.push_back(0);  vertex.push_back(1);  vertex.push_back(0);
vertex.push_back(0);  vertex.push_back(0);  vertex.push_back(1);

vertex.push_back(1);  vertex.push_back(1);  vertex.push_back(0);
vertex.push_back(1);  vertex.push_back(0);  vertex.push_back(1);
vertex.push_back(0);  vertex.push_back(1);  vertex.push_back(1);
vertex.push_back(1);  vertex.push_back(1);  vertex.push_back(1);
```

```
connectivity.push_back(0);
connectivity.push_back(1);
connectivity.push_back(4);
connectivity.push_back(2);

connectivity.push_back(1);
connectivity.push_back(5);
connectivity.push_back(7);
connectivity.push_back(4);

connectivity.push_back(3);
connectivity.push_back(6);
connectivity.push_back(7);
connectivity.push_back(5);

connectivity.push_back(2);
connectivity.push_back(6);
connectivity.push_back(3);
connectivity.push_back(0);

connectivity.push_back(2);
connectivity.push_back(4);
connectivity.push_back(7);
connectivity.push_back(6);

connectivity.push_back(0);
connectivity.push_back(3);
connectivity.push_back(5);
connectivity.push_back(1);

for(int k=0;k<vertex.size()/3;k++)
  {color.push_back(1.0); color.push_back(1.0); color.push_back(1.0);}

texture.push_back(Tx0); texture.push_back(Ty0);
texture.push_back(Tx1); texture.push_back(Ty1);
texture.push_back(Tx2); texture.push_back(Ty2);
texture.push_back(Tx3); texture.push_back(Ty3);
texture.push_back(Tx4); texture.push_back(Ty4);
texture.push_back(Tx5); texture.push_back(Ty5);
texture.push_back(Tx6); texture.push_back(Ty6);
texture.push_back(Tx7); texture.push_back(Ty7);
```