Examen du cours de Modélisation et Visualisation du solide

Jeudi 27 mars 2008
Durée: 3h, documents autorisés

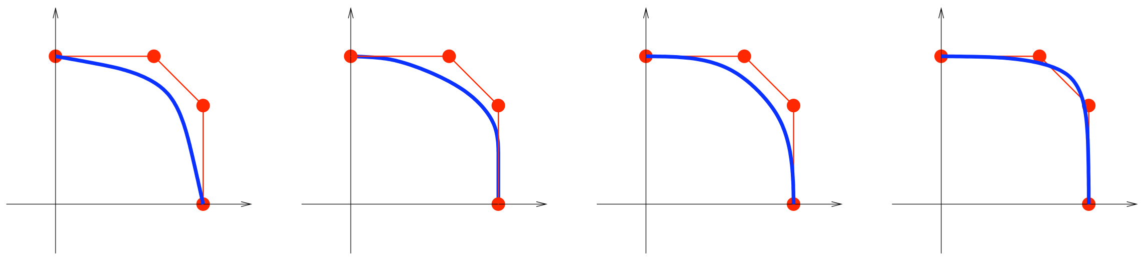
Les exercices sont indépendants, écrire les réponses sous chaque question sauf pour le dernier exercice.

Exercice 1. Perspective et splines

Question $\boxed{1}$ Soit $t \mapsto X(t)$ une courbe B-spline dans l'espace. On en prend une vue à partir d'une caméra située à distance finie. La courbe obtenue dans le plan est-elle une courbe de Bézier ? une courbe B-spline ? une NURBS ?

Question $\boxed{2}$ On suppose maintenant la caméra placée à distance infinie. La courbe obtenue dans le plan est-elle une courbe de Bézier ? une courbe B-spline ? une NURBS ?

Question $\boxed{3}$ Les 4 figures suivantes montrent le même polygone de contrôle, mais des courbes différentes. Sur quelle figure voit-on la courbe de Bézier de degré 3 associée au polygone donné ? Expliquer pour quel motif les autres peuvent être éliminées.



Question $\boxed{4}$ Soit $t \mapsto \gamma(t)$ une courbe de Bézier de degré k tracée dans un plan Π. Soit $P_0, P_1, \ldots, P_k$ son polygone de contrôle. Soit v un vecteur unitaire normal à Π. Soit $h > 0$ un réel. On appelle *cylindre* de hauteur h sur γ la surface balayée par la courbe γ lorsqu'on la translate d'une distance h dans la direction de v. On cherche a obtenir cette surface comme un carreau de Bézier. Quel est son bidegré ? Déterminer son réseau de contrôle.

Question $\boxed{5}$ On suppose désormais que γ est une courbe de Bézier rationnelle de degré k tracée dans Π, de polygone de contrôle $P_0, P_1, \ldots, P_k$ et de poids $w_0, w_1, \ldots, w_k$. On cherche a obtenir le cylindre de hauteur h sur γ comme un carreau de Bézier. Quels sont les poids à affecter aux points du réseau de contrôle ?

Exercice 2. Paraboloïde et équation implicite

On considère la surface quadrique $\mathcal{P}$ donnée par l'équation implicite :

$$z - x^2 - y^2 = 0 \ .$$

Question $\boxed{1}$ Quelle est la figure géométrique obtenue par intersection de $\mathcal{P}$ avec le plan d'équation $z = r$ pour $r \geq 0$? Et pour $r < 0$? $\mathcal{P}$ est-elle invariante par rotation autour d'un axe ? Si oui préciser lequel.

Question $\boxed{2}$ Déterminer (en fonction de x_0, y_0 et z_0) les coordonnées d'un vecteur normal à la surface $\mathcal{P}$ en un point (x_0, y_0, z_0) qui lui appartient.

Question $\boxed{3}$ Étant donnés deux réels u et v fixés, calculer le point d'intersection $M_{u,v}$ de $\mathcal{P}$ avec la droite $\Delta_{u,v}$ d'équation paramétrique :

$$\left\{ \begin{array}{c} x = u \\ y = v \\ z = -\lambda \end{array} \right. \quad , \quad \lambda \in \mathbb{R} \ .$$

En déduire, à partir de l'expression calculée à la question $\boxed{2}$, les coordonnées d'un vecteur normal $\vec{n}$ à $\mathcal{P}$ au point $M_{u,v}$.

Question $\boxed{4}$ On suppose que $\mathcal{P}$ est un objet réfléchissant. Un rayon lumineux est émis depuis une source située à l'infini sur l'axe Oz, en suivant la droite $\Delta_{u,v}$ selon la direction $\vec{i} = (0, 0, -1)$. En utilisant le fait que la direction de réflection $\vec{r}$ au point d'intersection $M_{u,v}$ est donnée par l'expression :

$$\vec{r} = \vec{i} - \frac{2\vec{n}}{\vec{n} \cdot \vec{i}}$$

calculer $\vec{r}$, et déterminer une équation paramétrique de la droite $\Delta'_{u,v}$ sur laquelle se trouve le rayon réfléchi.

Question $\boxed{5}$ Vérifier que le point de coordonnées $F = (0, 0, \frac{1}{4})$ appartient toujours à la droite $\Delta'_{u,v}$, quelles que soient les valeurs de u et v. On veut représenter l'objet $\mathcal{P}$ en utilisant une caméra placée en F et une source lumineuse placée à l'infini dans la direction Oz. Que peut-on dire de l'éclairement spéculaire de l'objet vu depuis cette caméra ? Réciproquement, on place la caméra à l'infini, pointée dans la direction $(0, 0, -1)$, et une source lumineuse en F. Expliquer, sans calculs supplémentaires, pourquoi on observe encore le même phénomène.

Exercice 3. OpenGL

Question $\boxed{1}$ Écrire un programme OpenGL / GLUT qui crée une fenêtre et affiche le triangle ABC dans le plan d'équation $z = 0$ sur un écran correspondant au pavé $[-1, 1]^2$, avec :

- A de coordonnées $(-0.5, 0)$ et de couleur rouge ;
- B de coordonnées $(0, 0.5)$ et de couleur verte ;
- C de coordonnées $(0.5, -0.5)$ et de couleur bleue.

On pourra se contenter de ne définir que le *callback* GLUT de réaffichage de la fenêtre.

Question $\boxed{2}$ Écrire un programme OpenGL / GLUT qui crée une fenêtre avec un tampon de profondeur associé, et qui effectue les tâches suivantes :

- activer :
  - le test du tampon de profondeur ;
  - les calculs d'éclairement et la lampe n°0 ;
  - le modèle d'éclairement simplifié, à l'aide de :
    `glEnable(GL_COLOR_MATERIAL);`
  - le modèle d'éclairement des deux côtés, à l'aide de :
    `glLightModeli(GL_LIGHT_MODEL_TWO_SIDE,1);`
  - la renormalisation automatique des vecteurs normaux non unitaires, à l'aide de :
    `glEnable(GL_NORMALIZE);`

–définir un *callback* GLUT de redimensionnement qui définit un *viewport* recouvrant la fenêtre, et une matrice de projection correspondant à une caméra perspective d'angle 50°, de mêmes proportions que la fenêtre, de distance minimale 1 et de distance maximale 10 ;

–définir un *callback* GLUT d'oisiveté qui force la fenêtre à se réafficher dès que possible ;

–définir un *callback* GLUT d'affichage qui place la caméra au point $(0, 0, 4)$, regardant vers le point $(0, 0, 0)$ et d'orientation verticale $(0, 1, 0)$.

En posant $\theta =$ `0.01*glutGet(GLUT_ELAPSED_TIME)`, appliquer à la scène successivement les 3 rotations d'angles 2θ, 3θ et 5θ autour respectivement des vecteurs $(1, 0, 0)$, $(0, 1, 0)$ et $(0, 0, 1)$.

On suppose que les constantes $N \in \mathbb{N}$, $r_1 > 0$ et $r_2 > 0$ sont données. Pour $0 \leq k \leq N$ on définit :

$$u = r_2 \cos \frac{5k\pi}{N} \quad , \quad v = r_2 \sin \frac{5k\pi}{N} \quad , \quad u_1 = r_1 - u \quad , \quad u_2 = r_1 + u$$

$$c = \cos \frac{2k\pi}{N} \quad , \quad s = sin \frac{2k\pi}{N}$$

$$P_1(k) = (cu_1, su_1, -v) \quad , \quad P_2(k) = (cu_2, su_2, +v) \quad \text{et} \quad \vec{n}(k) = (-cv, -sv, u) \ .$$

Dans ces conditions, afficher la bande de quadrilatères dont les deux bords sont définis par $P_1(k)$ et $P_2(k)$ (pour $0 \leq k \leq N$) et dont un vecteur normal est $\vec{n}(k)$ sur chacun des bords. On utilisera les valeurs constantes suivantes : $N = 1000$, $r_1 = 1$, $r_2 = 0.3$ et $\pi \approx 3.14159$;

Rappel : pour pouvoir utiliser les fonctions trigonométriques sin et cos dans votre programme il faudra penser à inclure `math.h`.