

Maîtrise d'Informatique

Examen d'Infographie II

3 juin 2004 - 3h

Michel Beaudouin-Lafon – mbl@lri.fr

Documents autorisés : 1 feuille A4 de notes personnelles.

Lisez l'ensemble de l'énoncé. Soyez clair, précis, concis. Justifiez vos réponses. Relisez-vous.

Problème – Calcul de silhouettes (15 points)

On considère des maillages de polygones représentés par la structure “half-edge” vue en cours et rappelée ci-dessous. Dans tout l'exercice, on ne considère que des maillages fermés, correspondant à la frontière de volumes simples (sans trous et qui ne s'auto-intersectent pas).

```
class Sommet {
    Point p;           // Les coordonnées du sommet
    Vector normal;    // Les coordonnées de la normale au sommet
    DemiArete a;      // Une arete qui contient ce sommet
}

class DemiArete {
    Sommet s;         // Le sommet initial de la demi-arête
    DemiArete autre; // L'arête "sœur" de cette demi-arête
    Face f;          // La face à laquelle appartient cette demi-arête
    DemiArete suiv;  // L'arête suivante de la face dans l'ordre de parcours
    Bool marque;     // Pour chaque paire de demi-arêtes,
                    // l'une est marquée vrai et l'autre faux
}

class Face {
    DemiArete a;      // Une demi-arête qui appartient à la face
    Vector normal;    // Les coordonnées de la normale de la face
}

class Maillage {
    listof Sommet sommets; // La liste des sommets
    listof DemiArete aretes; // La liste des demi-arêtes
    listof Face faces;     // La liste des faces
}
```

Pour faciliter l'écriture, on introduit le type “listof x” qui permet de manipuler des listes d'objets avec les opérations d'insertion, de destruction et de parcours suivantes :

```
lst.insert (e); // insérer l'élément e en tête de la liste lst
lst.append (e); // ajouter l'élément e à la fin de la liste lst
lst.remove (e); // retirer l'élément e de la liste lst
foreach (e, lst) { // parcourir les éléments de la liste lst
    // traiter l'élément e
}
```

Dans toutes les questions suivantes, on pourra ajouter des champs à la structure de données ci-dessus à condition de décrire et de justifier ces ajouts. Les programmes peuvent être écrits en Java, C ou C++ et doivent être commentés : c'est l'algorithme, plutôt que la syntaxe, qui est important.

Les 3 parties sont indépendantes.

Partie A (7 points)

1. Ecrire les algorithmes calculant les 9 relations d'adjacence entre sommets, arêtes et faces.
2. Ecrire les algorithmes permettant (a) d'insérer un sommet au milieu d'une arête et (b) d'insérer une arête entre deux sommets d'une face (figure 1).

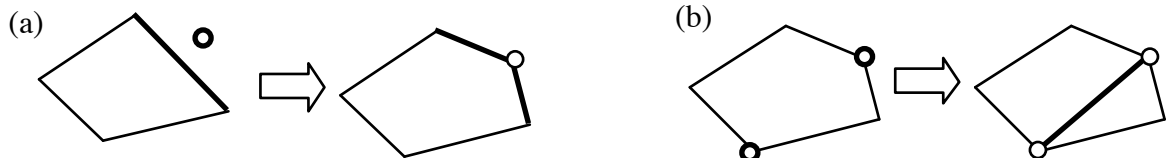


Figure 1 : (a) insertion d'un sommet. (b) insertion d'une arête.

Partie B (4 points)

On souhaite afficher la *silhouette* de l'objet représenté par le maillage. Etant donné une projection parallèle de l'objet, définie par le vecteur d , la silhouette est définie comme l'ensemble des arêtes adjacentes à deux faces f_1 et f_2 telles que f_1 est visible et f_2 cachée ou l'inverse (figure 2). Si f_1 et f_2 ont pour normales respectives n_1 et n_2 , cette condition se traduit par le fait que les produits scalaires $d \cdot n_1$ et $d \cdot n_2$ sont de signes opposés. La silhouette d'un objet peut être constituée de un ou plusieurs contours.

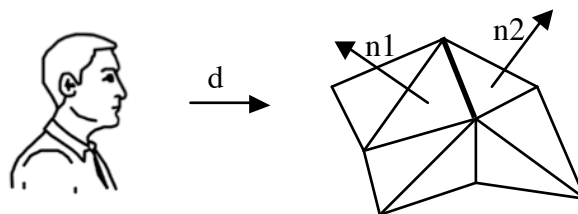


Figure 2 : l'arête en gras fait partie de la silhouette de l'objet.

3. Ecrire un algorithme qui calcule la liste des arêtes de la silhouette d'un objet. Pour un maillage ayant n faces et p arêtes, estimer le nombre d'additions, de multiplications et de comparaisons de cet algorithme.
4. Afin d'optimiser l'algorithme précédent, on commence par calculer, pour chaque face, le signe du produit scalaire $d \cdot n_i$ et on le stocke dans la structure de données, puis on calcule la liste des arêtes faisant partie de la silhouette. Ecrire ce nouvel algorithme, estimer sa complexité et la comparer à celle de la question 3.

Partie C (4 points)

Les algorithmes ci-dessus ne donnent pas de très bons résultats lorsque le maillage correspond à une surface courbe (spline ou NURBS par exemple). Dans ce cas, on préfère un algorithme qui utilise les normales aux sommets : soit une arête a de sommets s_1 et s_2 de normales respectives n_1 et n_2 . Si les produits scalaires $d \cdot n_1$ et $d \cdot n_2$ sont de signes opposés, la silhouette passe par un point p de cette arête (figure 3). On peut prendre le milieu du segment $s_1 s_2$ ou, mieux, calculer le point p par interpolation linéaire des produits scalaires afin de mieux approcher le point tel que $d \cdot n_p = 0$.

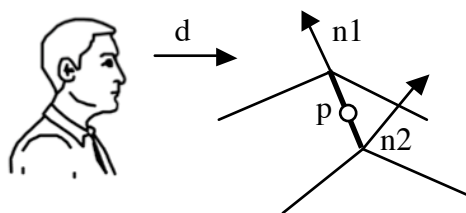


Figure 3 : le point p intersecte la silhouette.

Une fois calculés les points d'intersections de la silhouette avec les arêtes, on calcule les segments qui forment la silhouette de la façon suivante : si deux arêtes adjacentes intersectent la silhouette aux points p_1 et p_2 , calculés par la procédure ci-dessus, alors le segment $p_1 p_2$ fait partie de la silhouette (figure 4).

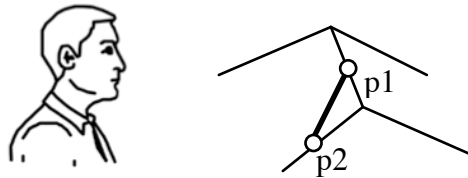


Figure 4 : le segment $p_1 p_2$ appartient à la silhouette.

5. Ecrire l'algorithme calculant la liste des segments constituant la silhouette. Il est suggéré de calculer d'abord les points d'intersection, de les stocker dans la structure du maillage, puis de déterminer les segments de la silhouette.
6. Modifier l'algorithme de la question 5 pour ajouter les contours de la silhouette au maillage, à l'aide des fonctions de la question 2.

Exercice – Traitement d'image (5 points)

On dispose de deux photos numériques A et B d'une même scène. Dans la photo A, les objets proches de l'objectif sont nets mais les autres sont flous. Dans la photo B, c'est l'inverse. On veut composer les deux images pour produire automatiquement une image nette partout.

1. Quelle est la différence entre le spectre d'une image nette et celui d'une image floue ? Que peut-on en déduire quand à l'effet d'un filtre passe-bas sur une partie floue d'une image par rapport à l'effet du même filtre sur une partie nette ?
2. En déduire une méthode qui produit une image binaire dont chaque pixel vaut 0 si l'image A est plus nette que l'image B en ce point, 1 si l'image B est plus nette que l'image A en ce point. En déduire un algorithme de composition des deux images produisant l'effet recherché.