



Rémy Malgouyres,
Algorithmes pour la synthèse d'images
et l'animation 3D,
DUNOD, 2002

TP n° 9 : Surfaces B -splines et déformation

Objectifs :

Le but de ce TP est d'implémenter une classe, héritant de la classe surface définie au TP précédent, permettant de représenter des surfaces B -splines bicubiques. Le sujet propose un travail facultatif consistant à permettre à l'utilisateur de déformer les surfaces.

1 Implémentation des surfaces B -splines

On propose de faire les calculs des points d'une surface B -spline bicubique par la méthode matricielle.

Pour la construction d'une surface B -spline cubique Q , de points de contrôle $P_{i,j}$, pour $i = 0, \dots, n - 1$ et $j = 0, \dots, m - 1$, on implémentera une fonction `precalculmatrices`, qui précalculera les produits :

$$M_{BS} \cdot G_{i,j} \cdot M_{BS}^T$$

où les matrices M_{BS} et $G_{i,j}$ sont définies dans le cours. On mémorisera les matrices obtenues dans des tableaux à deux dimensions de matrices.

Dans une fonction `getpoint`, on utilisera les matrices précalculées pour calculer un point $Q(s, t)$, étant donnés s et t en paramètre.

2 Déformation de surfaces

On ajoutera à la classe `Surface` trois fonctions membres :

- La fonction `void affichectrlpoints(CDC& cdc)` qui affiche les points de contrôle.
- La fonction `void GetCtrlpointSelect(Pixel pos, int & iproche, int & jproche)` qui retourne le numéro du point de contrôle dont la projection sur l'écran est la plus proche du pixel `pos`.
- `virtual void DeplaceCtrlpoint(int i, int j, Point3D v, CDC &cdc)` qui déplace le point de contrôle $P_{i,j}$ du vecteur v et rafraîchit la vue.

On créera ensuite les événements de sélection d'un point de contrôle par l'utilisateur, et le déplacement du point de contrôle à la souris.