COMPUTER GRAPHICS: HOMEWORK #5
DUE 2/12/08

1. Introduction: Shading, Spline Interpolation and Bump Maps.

A common technique for giving texture to images and polygons is based on changing the normals to the surface, according to a predetermined mapping function. In this homework, you will be asked to keep you image on a planar plate, but let the user control the “bumps” on the image. The assignment consists of several parts.

1.1. **Input and Display.** Your display window should be divided into 2 viewports. In the upper viewport, display the image plate `pic.jpg`. We denote by $II[u, v]$ the $[u, v]$ pixel of the Input image. We would assume that the width of this image is 512 pixels. Display the image using an orthographic projection, where the image is in the plane z=-10, and assume there is a source of light white at $(0, 512, 0)$. The function $F[u, v]$, whose details are described below, defines a displacement of the normal at each pixel for shading. The intensity of the light at pixel $II[u, v]$ is determined as in HW2. So intensity coming from different pixels varies according to the displaced normals given by F and the direction of the light.

The lower viewport of the display window shows the curve C, through which the user can control C.

1.2. **The curve $C(u)$ and the function $F(u, v)$.** The curve C, controlled by the user, is the concatenation of 4 curves $C_1(u) \dots C_4(u)$. They are determined via cubic splines (as defined in the slides). This curve C is the concatenation of 4 cubics C_1, C_2, C_3, C_4, where

- $C_1(0) = P_0,\ \ C_4(511) = P_4,$
- $C_1'(0) = C_1''(0) = 0$
- $C_1(127) = C_2(127) = P_1$; $C_2(255) = C_3(255) = P_2$; $C_3(384) = C_4(384) = P_3$
- $C_1'(127) = C_2'(127)$; $C_2'(255) = C_3'(255)$; $C_3'(384) = C_4'(384)$;
- $C_1''(127) = C_2''(127)$; $C_2''(255) = C_3''(255)$; $C_3''(384) = C_4''(384)$;

To compute this curve, you should use Hermite bases.

(2 is specified by 5 control points $(P_1 \dots P_5)$. Initially $P_i = ((i-1)128, 0)$, but the user can increase/decrease its y-value using the up and down arrows. So C is always smooth (that is, $C'(u)$ is continuous), and in addition, $C_1''(0)$ always equals 0.

The user should be allowed to select an individual control point for manipulation by using the left and right arrows to move along the splines. On each press of the $*$ key, the value of active control point $P_i = (u_i, v_i)$ is replaced by $(u_i, 2v_i)$. This allows the spline curvature to be exaggerated very quickly. Hitting the 'ESC' button resets each P_i to its original value.

1.3. **The bumps map.** The idea is to specify the normal to the plate as follows: We define the function $F(u,v) = (C(u), 0)$, and the normal $F(u,v)$ is approximated as the normal to plane passing through the points

$$(u, v, F(u,v)),\ (u+1, v, F(u+1, v)) \text{ and } (u+1, v+1, F(u+1, v+1))\ .$$

You will need a formula here to help you compute the normal from this data, and don't forget that the length of the normal vector is always 1.