CSC418/2504 Computer Graphics, Winter 2010

Assignment 2(b)

Part B (theory): Due 11:00am on Thursday 4 March 2010, at BA5266.

Interpolation using Overhauser/Catmull-Rom Splines

In part A of this assignment, you are creating a simple keyframe animation tool, a component of which involves so-called cubic Catmull-Rom spline interpolation. You are welcome to look up references on interpolation, but this discussion will be self-contained.

We have seen that the parametric representation of objects affords a way to characterise behaviour in space through mapping every point in a parametric domain, usually consisting of a unit interval, unit square or unit cube, to a desired object. For example, we can write the line segment between two points $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{R}^n$ parametrically as

$$\begin{aligned} P(t) &= (1-t)\mathbf{p}_1 + t\mathbf{p}_2 \\ &= b_1(t)\mathbf{p}_1 + b_2(t)\mathbf{p}_2 \\ &= \sum_{i=1}^{2} b_i(t)\mathbf{p}_i \end{aligned}$$

for $t \in [0,1]$. Now, in what might seem like notational overkill, the *basis functions* $b_1(t) = 1-t$ and $b_2(t) = t$ can be seen as blending the values of each component of the points $\mathbf{p}_1$ and $\mathbf{p}_2$. Regardless of the dimension of the space in which these points lie, for a given value of t, the $b_i(t)$ act to give a weighted sum of the relevant points. We can thus substitute values of t to understand how the blending operates. For example, setting t to 0, $\frac{1}{2}$, and 1 in $P(t)$ gives us exactly what we would expect, namely that these values respectively interpolate $\mathbf{p}_1$, the midpoint of the line segment, and $\mathbf{p}_2$. Note that the degree d of the interpolation, in this case linear or $d = 1$, means that there will be $d+1$ basis functions. Thus if d is the degree, then the *order* of the basis functions is $d+1$. In a general interpolation problem of degree $d-1$, we would thus have basis functions $b_1, b_2(t), \cdots, b_d(t)$, and the blending operation would look like

$$P(t) = \sum_{i=1}^{d} b_i(t)\mathbf{p}_i.$$

One example of basis functions would be $b_i(t) = t^i$, in other words, the usual polynomial basis, but it is not very useful for interpolation. General interpolation problems require that we construct a curve that passes through a sequence of points $[\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n]$. While there are ways to interpolate all these points simultaneously using n basis functions of degree $d = n-1$, these techniques are generally neither practical nor even desirable. A *piecewise* interpolation scheme breaks the overall interpolation problem up into a sequence of interpolations over pieces of functions, often called *splines*. This term harkens to a time when wooden or metal splines were shaped under pressure and then attached to give a complex curve from simpler pieces. The obvious way to do piecewise linear interpolation, for example, would be to draw line segments between each $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, for $i = 1, 2, \cdots, n-1$. Thus the result of interpolating n points gives us $n-1$ line segments $P_i(t)$, with

$$P_i(t) = b_1(t)\mathbf{p}_i + b_2(t)\mathbf{p}_{i+1},$$

for $i = 1, 2, \cdots, n-1$. Notice that the end-point of one piece becomes the start-point of the next. A higher degree piecewise interpolating spline might be desirable when we want the resulting interpolating curve to be smooth at the points at which the curve segments are connected. This requires that we construct basis functions that do more than merely interpolate values.

The uniform cubic Overhauser or Catmull-Rom spline interpolation uses basis functions of degree three, and thus there are four of them:

$$\begin{aligned}
b_1(t) &= \frac{1}{2}(-t^3 + 2t^2 - t) \\
b_2(t) &= \frac{1}{2}(3t^3 - 5t^2 + 2) \\
b_3(t) &= \frac{1}{2}(-3t^3 + 4t^2 + t) \\
b_4(t) &= \frac{1}{2}(t^3 - t^2)
\end{aligned}$$

for $t \in [0, 1]$. Such a basis function definition lends itself naturally to a matrix representation, as we shall see later in the course, but we will stick with the summation form. On a sequence $[\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n]$ of points, each curve segment $P_k(t)$ for $k = 1, 2, \cdots, m$ is defined as:

$$P_k(t) = \sum_{i=1}^{4} b_i(t)\mathbf{p}_{k-1+i}. \tag{1}$$

Thus, for example, the first curve segment $P_1(t)$ looks like this:

$$P_1(t) = \sum_{i=1}^{4} b_i(t)\mathbf{p}_i. \tag{2}$$

$P_2(t)$ would involve dropping $\mathbf{p}_1$ and incorporating $\mathbf{p}_4$. Please note that these splines are slightly different from those found in the notes, and the code in your assignment takes on a different form of them. This is the classical form.

And now, to the questions!

1. What is m above in terms of n? In other words, how many curve segments are generated by the above scheme on n data points?

2. For each curve segment $P_k(t)$, which points are interpolated? Hint, start with $P_1(t)$ and explore the $b_i(t)$ for well-chosen values of t. Then generalise to curve segment P_k.

3. These basis functions were designed to do more than interpolate. Write down the derivative $\dot{P}_k(t)$, namely the derivative of $P_k(t)$ with respect to t. If you have difficulty, do this first for linear interpolation. Write down a basis function representation for $\dot{P}_k(t)$ in the form of Eq. 1.

4. Using Question 3 or your own derivation, give the value of the tangent of each $P_k(t)$ both at left and right endpoints (i.e., when t is 0 or 1). Show that $\dot{P}_{k-1}(1) = \dot{P}_k(0)$, for $k = 1, \cdots, m$.

   For the next two questions, remember that any space curve segment $P_k(t)$ is actually a tuple of curve segments $(x_k(t), y_k(t), z_k(t), \cdots)$ depending on the dimensionality of the points $\mathbf{p}$ to be interpolated. Most of the operations we perform on P_k can thus be reduced to performing the same operation on each of the parametric components of P_k.

5. Let us consider a curve segment $P(t) = P_1(t)$, which is defined on the sequence of points $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4]$ as in Eq. 2. As in Assignment 1, let $\mathbf{T}$ be a linear transformation (which we can think of as all affine transformations that do not have a translation component). Show that

$$\mathbf{T}P_1(t) \quad = \quad \sum_{i=1}^{4} b_i(t)\mathbf{T}\mathbf{p}_i.$$

This is easy! Thus we can rotate and zoom the curve segments merely by performing the operation on the points to be interpolated.

6. Now, let's suppose we merely wish to pick up the curve and move it around. Let's restrict the $\mathbf{p}_i$ to be in $\mathbb{R}^2$ (this is just for notational convenience, since as I've said before the result is independent of dimension). We thus wish to translate the entire curve by (say) an offset of $(\Delta x, \Delta y)$. As in the previous question, we would like to be able to translate the curve merely by translating the original points and then performing the interpolation. What must be true of the basis functions for the following to be true:

$$P_1(t) + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad = \quad \sum_{i=1}^{4} b_i(t) \left(\mathbf{p}_i + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right).$$

Show that the Catmull-Rom basis functions have this property. Does the line segment basis from above have this property as well?

The two questions, taken together, show that the Catmull-Rom basis is invariant to all affine transformations. Indeed the entire question could be formulated using the homogeneous matrix representations we have employed in class for affine transformations.

Bonus: Is the Catmull-Rom basis invariant under perspective projection? Hint: set up the pinhole viewing schema and derive the perspective projection for, say, $P_1(t) = (x_1(t), y_1(t), z_1(t))$. If P_1 is invariant under perspective, prove it. If you think it isn't, then give a specific set of points that yield a counter-example.

7. Suppose we wish to compute all the intersections of a line or ray with a Catmull-Rom curve segment. Let's restrict ourselves to two-dimensional lines and curves. Specifically, suppose the ray passes through two points $A(a_x, a_y)$ and $B(b_x, b_y)$ in $\mathbb{R}^2$. Define the line parametrically by $L(t) = A + s(B - A)$ for $s \in \mathbb{R}$, which you will then intersect with a Catmull-Rom curve, say $P_1(t)$ as above. Your goal is to solve for all values of s such that $L(s)$ lies on a Catmull-Rom curve $P(t)$. Why is this not easy to do? How many such s can there be? How few can there be? Now, think of all the invariance properties we have discovered in this assignment and the affine transformations we have taken in class; then think about how you might go about solving this problem by reducing it to an easier problem. The more detail you give in your algorithm to solve this, the better your grade, but you can stick to a written sketch or some pseudocode.