CMSC 23700	**Introduction to Computer Graphics**	**Homework 3**
Winter 2012		**Assigned March 6**
		Due 6pm March 13

In class we derived the Hermite spline and its basis functions. That is, the cubic polynomial $s(t)$ for which $s(0) = p_0$, $s'(0) = m_0$, $s(1) = p_1$, and $s'(1) = m_1$ is

$$s(t) = p_0 + m_0 t + (-3p_0 - 2m_0 + 3p_1 - m_1)t^2 + (2p_0 + m_0 - 2p_1 + m_1)t^3;\ t \in [0, 1] \tag{1}$$

We also found that this could be re-written as:

$$s(t) = p_0 b_0(t) + m_0 b_1(t) + p_1 b_2(t) + m_1 b_3(t);\ t \in [0, 1] \tag{2}$$

$$b_0(t) = 1 - 3t^2 + 2t^3 \qquad (b_0(0) = 1) \tag{3}$$

$$b_1(t) = t - 2t^2 + t^3 \qquad (b_1'(0) = 1) \tag{4}$$

$$b_2(t) = 3t^2 - 2t^3 \qquad (b_2(1) = 1) \tag{5}$$

$$b_3(t) = -t^2 + t^3 \qquad (b_3'(1) = 1) \tag{6}$$

which expresses the Hermite spline as a linear combination of basis functions $b_i(t)$, each one scaled by either the position (or value) controls p_0, p_1 or the slope (or derivative) controls m_0, m_1. By design, each of the basis functions is zero for all but one of the values or derivatives at the endpoints ($t = 0$ and $t = 1$), as listed above. **See the Feb 28 lecture notes for graphs of the b_i functions**, or you can plot these yourself with some function plotting program.

Consider a sequence of (scalar) values $p[n]$ defined at the integers n. As a discrete function, p does not have continuous derivatives, but we can learn a slope $m[n]$ of p around n from the neighboring two values:

$$m[n] = \frac{p[n+1] - p[n-1]}{2} \tag{7}$$

We want to arrive at a way of blending between the $p[n]$ control points with some continuous function $f(x)$ that interpolates through them: $f(n) = p[n]$ for all n.

1. Consider the interval $x \in [n, n+1]$, and the Hermite spline $f(x)$ within it defined by $f(n) = p[n]$, $f'(n) = m[n]$, $f(n+1) = p[n+1]$, $f'(n+1) = m[n+1]$. By expanding (7) in (2), find the 4×4 matrix M such that for $x \in [n, n+1]$,

$$f(x) = \begin{bmatrix} b_0(x-n) & b_1(x-n) & b_2(x-n) & b_3(x-n) \end{bmatrix} M \begin{bmatrix} p[n-1] \\ p[n] \\ p[n+1] \\ p[n+2] \end{bmatrix}$$

   Finding the coefficients in M (only 0, 1, $\pm\frac{1}{2}$) is *not* helped by expanding the formulae for the b_i. Note that if $t = x - n$ then $x \in [n, n+1] \Leftrightarrow t \in [0, 1]$.

2. By re-writing the product of the b_i row vector and matrix M above, find expressions (in terms of the b_i) for the four $h_{[l,h]}(t)$ functions which satisfy (for $x \in [n, n+1]$)

$$f(x) = \begin{bmatrix} h_{[1,2]}(x-(n-1)) & h_{[0,1]}(x-n) & h_{[-1,0]}(x-(n+1)) & h_{[-2,-1]}(x-(n+2)) \end{bmatrix} \begin{bmatrix} p[n-1] \\ p[n] \\ p[n+1] \\ p[n+2] \end{bmatrix}.$$

Note that if $t = x - n$ and $x \in [n, n+1]$, then the interval $[l, h]$ in the subscript of $h_{[l,h]}(t)$ gives the range of t over which $h_{[l,h]}(t)$ may be evaluated.

3. Putting together the results above, compose a definition of a single function $h(t)$ in terms of the b_i that satisifies (for $x \in [n, n+1]$):

$$f(x) = \sum_{i=-1}^{2} p[n+i]h(x - (n+i)) \tag{8}$$

   The definition should specify the value of $h(t)$ for $t < -2$, $t \in [-2, -1]$, $t \in [-1, 0]$, $t \in [0, 1]$, $t \in [1, 2]$, and $t > 2$. **Also, make a sketch of your** $h(t)$ by looking at the plots of the $b_i(t)$ basis functions and considering the combinations of $b_i(t)$ involved in each interval of $h(t)$. Hint: do a Google image search for "plot Catmull-Rom filter" (without the quotes).

4. Show that Eq (8) can be re-written as:

$$f(x) = \sum_{i=-\infty}^{\infty} p[i]h(x - i). \tag{9}$$

   At most, how many terms in this summation will be non-zero? Eq. (9) is in fact the formula for a convolution of discrete data $p[i]$ with a continuous function $h(x)$, which in our case is the Catmull-Rom function you have derived. Note that in this convolution, there is a **single** basis function $h(t)$, called the *kernel* or the *filter*, copies of which are being scaled by the discrete values $p[i]$. The original Hermite spline, in contrast, involves four different basis functions $b_i(t)$.

Wikipedia describes convolution at http://en.wikipedia.org/wiki/Convolution, but doesn't cover the specific case of discrete data and continuous kernel. Figures 5.23, 5.24, and 5.25 on pages 121 and 122 of Real-Time Rendering do illustrate this kind of convolution, though unfortunately without any formulae.

The steps above essentially amount to some book-keeping and re-writing (but *not* lots of algebraic transformations or lengthy expressions), with the goal of conceptually connecting two superficially different ways of blending values: splines and filters.