Computer Graphics	Prof. Brian Curless
CSE 557	Winter 2003

Homework #2

Shading, parametric curves, subdivision curves, final project selection

Assigned: Friday, February 28, 2002

Due: Thursday, March 6, 2002
(at the beginning of class)

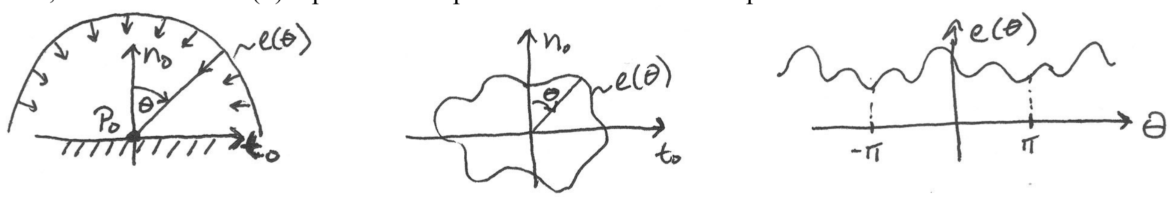
Directions: Please provide short written answers to the following questions. Feel free to talk over the problems in general terms with classmates, but please *answer the questions on your own*.

Name:____

1. Shading as signal processing

In this problem, we will consider the effect of diffuse shading in a signal processing framework. To make the analysis tractable, we will work in a 2D world ("flatland"), but the concepts generalize to 3D. We will also assume a monochrome world, i.e., light intensities and reflection coefficients will be scalar values, not (*r,g,b*) triples. We will analyze the problem in a local illumination framework, i.e., no interreflections or shadows.

Consider a point P_0 on a diffuse surface with normal $\mathbf{n}_0$ and diffuse coefficient k_d, sometimes written ρ and called the "albedo." This surface is illuminated by a distant and arbitrarily complex environment. The environment illumination is described by the function $e(\theta)$. The figure below illustrates this illumination relative to the surface point, and then shows $e(\theta)$ represented in polar coordinates and as a periodic function over the real line.



Note that θ is measured from the surface normal $\mathbf{n}_0$ and runs clockwise in this problem (toward tangent $\mathbf{t}_0$). The light reflected from this point is an integral of diffuse reflection over all incoming light directions:

$$l = \int_{-\pi/2}^{\pi/2} \rho\, e(\theta)\cos(\theta)d\theta$$

We could compute this integral by sampling over the lighting directions, weighting by $\cos\theta$, summing the result, and dividing by the number of samples. The goal of this problem is to find an approximate method that gives reasonably accurate results with far less computation.

Before proceeding with the problem, we note a few theorems from Fourier analysis. The Fourier transform of a 1D signal is:

$$\Im_{1\mathrm{D}}\{f(\theta)\} = F(\phi) = \int_{-\infty}^{\infty} f(\theta)e^{-i2\pi\phi\theta}d\theta$$

Note that the variables θ and ϕ have been substituted for *x* and *s* for this problem. 1D continuous convolution can be written as:

$$f(\theta) * h(\theta) = \int_{-\infty}^{\infty} f(\theta')h(\theta-\theta')d\theta'$$

Some useful Fourier transform pairs:

$$\underset{\Im^{-1}}{\overset{\Im}{\rightleftarrows}}$$

$f(\theta) * h(\theta)$	$F(\phi)H(\phi)$
$f(a\theta)$	$\frac{1}{\lvert a\rvert}F\left(\frac{\phi}{a}\right)$
$\Pi(\theta) = \begin{cases} 1 & \lvert\theta\rvert < 1/2 \\ 1/2 & \theta = 1/2 \\ 0 & \lvert\theta\rvert > 1/2 \end{cases}$	$\mathrm{sinc}(\phi) = \frac{\sin(\pi\phi)}{\pi\phi}$
$\mathrm{III}(\theta)$	$\mathrm{III}(\phi)$
$e^{j\theta}$	$\delta(\phi-\theta)$
$\cos(\theta) = \frac{e^{j\theta}+e^{-j\theta}}{2}$	$\frac{1}{2}\delta\left(\phi-\frac{1}{2\pi}\right)+\frac{1}{2}\delta\left(\phi+\frac{1}{2\pi}\right)$
$\sin(\theta) = \frac{e^{j\theta}-e^{-j\theta}}{2}$	$\frac{1}{2}\delta\left(\phi-\frac{1}{2\pi}\right)-\frac{1}{2}\delta\left(\phi+\frac{1}{2\pi}\right)$

1. Shading as signal processing (cont'd)

Scaling the argument of a delta function is equivalent to changing its area under integration. As a result:

$$\delta(a\theta - b) = \frac{1}{|a|}\delta\left(\theta - \frac{b}{a}\right)$$

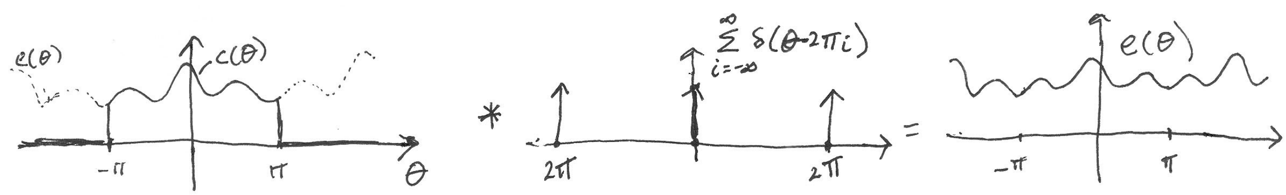
A corollary of this delta function behavior is:

$$\text{III}(a\theta) = \sum_{i=-\infty}^{i=\infty} \delta(ax - i) = \frac{1}{|a|}\sum_{i=-\infty}^{i=\infty} \delta\left(\theta - \frac{i}{a}\right)$$

a. First, we note that $e(\theta)$ is periodic. Periodic functions can be represented as a *Fourier series*, i.e., a set of evenly spaced samples (delta functions of various "heights") in the Fourier domain, where the location of each frequency sample is called a *harmonic*. Consider one period of $e(\theta)$, varying from $-\pi$ to π. We can select for this central component, call it $c(\theta)$, using the box function:

$$c(\theta) = \Pi\left(\frac{\theta}{2\pi}\right)e(\theta)$$

From $c(\theta)$, we can construct $e(\theta)$ using the impulse train and the shifting theorem of delta functions:



What is the Fourier transform, $E(\phi)$, of $e(\theta)$, in terms of $C(\phi)$?

b. Now consider a diffuse surface you want to shade with the environment $e(\theta)$. So far, we have represented the environment with respect to a particular normal, $\mathbf{n}_0$. Other surface normals will be rotated versions of this normal at an angle α with respect to $\mathbf{n}_0$. From the point of view of these normals, the environment appears to be rotated by α. Write out the equation for $l(\alpha)$, the light reflected from an arbitraray surface normal, in terms of the rotated environment, $e(\theta - \alpha)$, and modify the integrand so that the limits of integration are $+\infty$ and $-\infty$.

c. What is the Fourier transform, $L(\phi)$, of $l(\theta)$, in terms of $C(\phi)$? Your answer should take the form of another Fourier series.

d. The result of (c) shows that the Fourier series spectrum of $e(\theta)$ is attenuated at each harmonic. What is the attenuation weight for the i-th harmonic? Perform simplifications so that your answer contains no sines or cosines.

e. When shading diffuse surfaces with environment illumination, how would you save space in representing the environment illumination approximately? How would you save time in computing diffuse reflection from each point on the surface? Justify your answers.

f. Suppose we have a real world sphere (a circle in 2D for this problem)) with uniform albedo over the surface, and we take a sequence of images of the sphere (circle) so that we see how much light is reflected for every possible normal. R. W. Priesendorfer conjectured 25 years ago that one could take these observations and, in theory, reconstruct an arbitrary lighting environment completely. Was he right? Justify your answer.

2. Direct manipulation of B-spline curves

One of the shortcomings of the parametric curves we've described in class is the fact that the only ``handles" for modifying the curve are the control points. For approximating curves, such as B-splines the handles don't even lie on the curves. A nice feature would be to allow the user to pick a point on the curve, drag that point around, and have the curve follow (and interpolate) the point.

[Note: in this problem, we are only directly computing adjustments to the B-spline control points. The Bezier control points are derived from the B-spline control points and are free to move indirectly (i.e., you do not need to keep track of what happens to the Bezier control points that are not being moved directly as in part (a)).]

a. As shown in class, a set of four cubic B-spline control points (a.k.a., de Boor points), $B_0 ... B_3$ generate a set of cubic Bezier control points, $V_0 ... V_3$. Let's "associate" V_0 with B_1 (the tip of it's A-frame). Likewise, we can associate V_3 with B_2. Clearly, both V_0 and V_3 lie on the curve. If we were to select and move V_0 by ΔV_0, how would we compute ΔB_1 (the change in B_1) if the remaining B-spline control points remain fixed? [Aside: we could perform the same analysis if we were to move V_3 and update B_2 accordingly.]

b. Before improving on (a), recall that, cubic Bezier curves can be written in the form:

$$Q(u) = \sum_{i=0}^{3} V_i b_i(u)$$

where the $b_i(u)$ are the cubic Bezier-Bernstein polynomials.

Similarly, B-spline curves can be written as:

$$Q(u) = \sum_{i=0}^{3} B_i n_i(u)$$

where $n_i(u)$ are the B-spline basis functions. Solve for the polynomial forms of the cubic B-spline basis functions.

c. Now, let's say we pick an arbitrary point, $Q(u_o)$ on the curve, and then move it by $\Delta Q(u_o)$. We could move either B_1 or B_2 to accommodate, but a more natural choice is to move both of them in proportion to how close u_o is to 0 or 1. Let's impose the constraint that:

$$u_o \Delta B_1 = (1 - u_o) \Delta B_2$$

Compute ΔB_1 and ΔB_2 as a function of $\Delta Q(u_o)$. Your solution should contain B-spline basis functions, n_i, but you do not need to expand them into their polynomial forms when writing your answer.
What happens at u_o equal to 0 or 1?

d. Consider a curve built from many de Boor points. Let's say you have repositioned an arbitrary point on the curve using the method described in (c). If you now reposition a point in one of the immediately neighboring Bezier segments, will the point you originally repositioned still, in general, be interpolated? Justify your answer.

3. Subdivision curves

The subdivision mask for quadratic B-spline curves is:

$$r = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Perform the following steps in analyzing the fundamental characteristics of this subdivision curve:

a. Using a construction similar to the one we used in class, develop the subdivision matrix, **S**, for this curve. As part of your answer, sketch out a figure that illustrates the neighborhood you are analyzing.

b. Compute the eigenvalues and eigenvetors of **S**. Recall from linear algebra that eigenvalues are found to be roots of det(**S** – λ **I**) = 0, and the eigenvectors can be solved from the equation **S** v = λ v for each eigenvalue.

c. Solve for the limit mask of this subdivision scheme.

d. Solve for the tangent mask of this subdivision scheme.

The subdivision mask for the Dyn-Levin-Gregory (DLG) *interpolating* subdivision scheme is:

$$r = \frac{1}{16}(-2, 5, 10, 5, -2)$$

e. What is the subdivision matrix for the DLG subdivision scheme? Again, your answer should include a sketch of the neighborhood you are analyzing.

4. Final project

Send email to both Keith and myself describing who you plan to work with and what you plan to develop for your final project. Explain your division of labor, and describe the artifact you hope to produce. This email is due by midnight of March 6 (same due date as this rest of this homework). We will set up a meeting with each team to discuss their project shortly afterward. You will demo the project itself on the morning of Thursday, March 20, with a concise web page write-up (with artifacts) due by 5pm on Friday, March 21.