Higher-Order Basis Functions in Radiosity.

The Galerkin [Zatz], Finite Element [Troutman] and Wavelet [Gortler] (all SIGGRAPH '93) radiosity methods are all based on the use of higher-order elements, rather than the constant elements used by previous radiosity algorithms. Papers on these methods have concentrated on high-level matters, leaving the mechanics of dealing with function bases as an exercise to the reader. Here I present an quick, informal guide to using higher-order bases in a radiosity scheme. We'll take a look at projecting functions onto function spaces, and at how to carry out the fundamental shooting or gathering step common to all radiosity methods by using the ***matrix transport equation***.

Projection

We can represent the radiosity over a patch by a linear combination of basis functions. There are n functions $\boldsymbol{b}_i$, and the radiosity is represented by storing the coefficients of these functions in an n-element vector, $\underset{\sim}{x}$. The radiosity function over the patch is then just $\sum_i \boldsymbol{b}_i x_i$. An obvious question to ask is how we can convert (or *project*) an arbitrary function $\boldsymbol{f}$ over the patch into this representation.

Conceptually, we let $\boldsymbol{A}$ be the matrix containing $\boldsymbol{b}_i$ as columns. Note that as $\boldsymbol{b}_i$ are functions, we cannot explicitly form $\boldsymbol{A}$. We wish to find the vector $\underset{\sim}{x}$ such that $\boldsymbol{A}\underset{\sim}{x} = \boldsymbol{f}$. Normally we could simply invert $\boldsymbol{A}$ to do this, however, we cannot directly manipulate $\boldsymbol{A}$. We can only take the inner product between two functions, as follows:

$$\boldsymbol{f} \cdot \boldsymbol{g} = \int \boldsymbol{f}(x)\boldsymbol{g}(x)dx \tag{1}$$

We circumvent this problem by multiplying through by $\boldsymbol{A}^t$, yielding:

$$\begin{aligned} (\boldsymbol{A}^t\boldsymbol{A})\underset{\sim}{x} &= (\boldsymbol{A}^t\boldsymbol{f}) \\ D\underset{\sim}{x} &= \underset{\sim}{f} \end{aligned} \tag{2}$$

Now the two matrices in the equation are real; we have $D_{ij} = \boldsymbol{b}_i \cdot \boldsymbol{b}_j$ and $f_i = \boldsymbol{b}_i \cdot \boldsymbol{f}$, and we can solve directly for $\underset{\sim}{x}$. Thus the projection of a function onto a function space is found by taking its inner product with each of the basis functions, and multiplying the resulting vector by D^{-1}. This projection will not always be exact, as usually the basis doesn't span all possible functions. In fact, the process is analogous to least-squares fitting.

The inner-product integrals can be calculated by hand, by using a symbolic computation system such as Mathematica, or numerically, by using a *quadrature rule* of sufficient accuracy. We shall learn more about quadrature rules in a couple of paragraphs.

The Gathering or Shooting Step

So how does this help us with our radiosity program? The central operation in radiosity programs is a shoot or gather step, where the radiosity of one patch is used to calculate the resulting irradiance of another patch. We need to find a way to take the radiosity coefficients of the source patch and calculate coefficients for the destination patch that best match its irradiance. Conceptually this involves reconstructing the radiosity function over the source patch, integrating it over a kernel func-

tion to produce the irradiance on the destination patch, and then projecting that irradiance function into the receiving patch's basis. We can represent this process as:

$$d_j = \int_{y \in R} b_j \left(\sum_i \int_{x \in S} \boldsymbol{K}(x, y) \boldsymbol{b}_i(x) s_i dx \right) dy, \tag{3}$$

where $\boldsymbol{K}(x, y)$ is the standard radiosity kernel,

$$\boldsymbol{K}(x, y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2}, \tag{4}$$

and *s* and *d* are the source and destination coefficients.

A quadrature rule lets us estimate the integral of a function by sampling it at a set of points, x_i, and forming the weighted sum of these samples, with weights w_i. If we replace a function $\boldsymbol{f}$ we wish to integrate with the vector $\underset{\sim}{f} = [\boldsymbol{f}(x_i)]$, and form the vector $\underset{\sim}{w} = [w_i]$, the integral of $\boldsymbol{f}$ according to the quadrature rule is just $\underset{\sim}{w} \cdot \underset{\sim}{f}$. A quadrature rule is guaranteed to be exact under certain conditions; the commonly-used n-point gaussian quadrature rule, for example, will be exact if the integrand is a polynomial of order *2n-1* or less. Applying such a rule to the inner integral of equation (2) gives:

$$\sum_i \underset{\sim}{w} \cdot (s_i \underset{\sim}{b}_i \bullet \underset{\sim}{k}_k), \tag{5}$$

where $[a_i] \bullet [b_i] = [a_i b_i]$, and $\underset{\sim}{k}_k$ is the vector of kernel samples from all quadrature points on the source patch to quadrature point x_k on the receiving patch. We can rewrite this as $\underset{\sim}{k}_j^t M \underset{\sim}{s}$, where $M_{ij} = w_j B_{ij}$. If we perform the same process on the outer integral. We find that:

$$\underset{\sim}{d} = D^{-1} M^t K M \underset{\sim}{s} \tag{6}$$

where $K_{ij} = \boldsymbol{K}(x_i^r, x_j^s)$. This is the ***matrix transport equation***.

We can precalculate the matrices to the left and right of the kernel matrix in this equation. During the radiosity simulation, each time we form a new link between patches, we calculate *K* by sampling the kernel between each pair of quadrature points on the two patches, and then perform two matrix multiplies to get the transport matrix *T* for that link. From then on, each gather operation across that link is a single matrix multiply: $\underset{\sim}{d} = T \underset{\sim}{s}$.

We have assumed here that all patches share the same basis and use the same quadrature method. This is not always the case: we might use both quadrangles and triangles as elements in the simulation, or use different quadrature rules for some patches to address the singularity in $\boldsymbol{K}$. This is simply addressed: we form a different *M* for each basis and quadrature pair, and keep in mind that in the transport equation the matrices to the left of *K* (D^{-1} and M^t) are in terms of the basis and quadrature rule of the destination patch, and the matrix to the right (*M*) to the right is in terms of the basis and rule of the source patch.