Wavelet Radiosity – Part I

Summary of the article ‘Wavelet Radiosity’
by Gortler, Schröder, Cohen & Hanrahan,
parts 1 thru 6 and 9.

By Axel Hjälm & Rikard Marklund

Overview

Although radiosity methods generally are effective means to solving the global illumination problem for Lambertian surfaces, they are still time (and memory) consuming. Traditional radiosity approximates the radiosity integral by a finite set of basis functions, resulting in a set of linear equations to be solved. Typically, for a scene with *n* elements, a system of *n* linear equations has to be solved, involving the evaluation of some n^2 coefficient. Different methods can be used that exploits various features of the scene or the radiosity integral. The goal of these methods are to reduce the number of coefficients in the equation system, reducing the number of evaluation to *n*.

Wavelet radiosity offers a method of reducing the number of coefficient, resulting in relatively simple equation system to solve (i.e. we get a very sparse matrix of coefficients). This can be achieved by approximating the radiosity integral by a special set of functions called wavelets. The approximation is done by *projecting* the kernel of the integral onto a finite set of function, chosen so that the resulting coefficient are as small and few as possible.

The radiosity integral function

At the core of the radiosity integral is the so called *kernel:*

$$k(s_1, s_2, t_1, t_2) = \rho(s_1, s_2)\frac{\cos\Theta_s \cos\Theta t}{\pi_{st}^2}$$

The kernel describes the visibility relationship between two points in the domain. Where *r* is large relative to the distance between to two patches, the kernel is very well represented (approximated) by a low order polynomial. Exceptions to this are e.g. corners (where r^2 goes to zero and the kernel is singular) and shadow discontinuities.

The goal of wavelet radiosity is to approximate this function in a precise and efficient way. In order to do that, it needs to be projected on a set of basis functions.

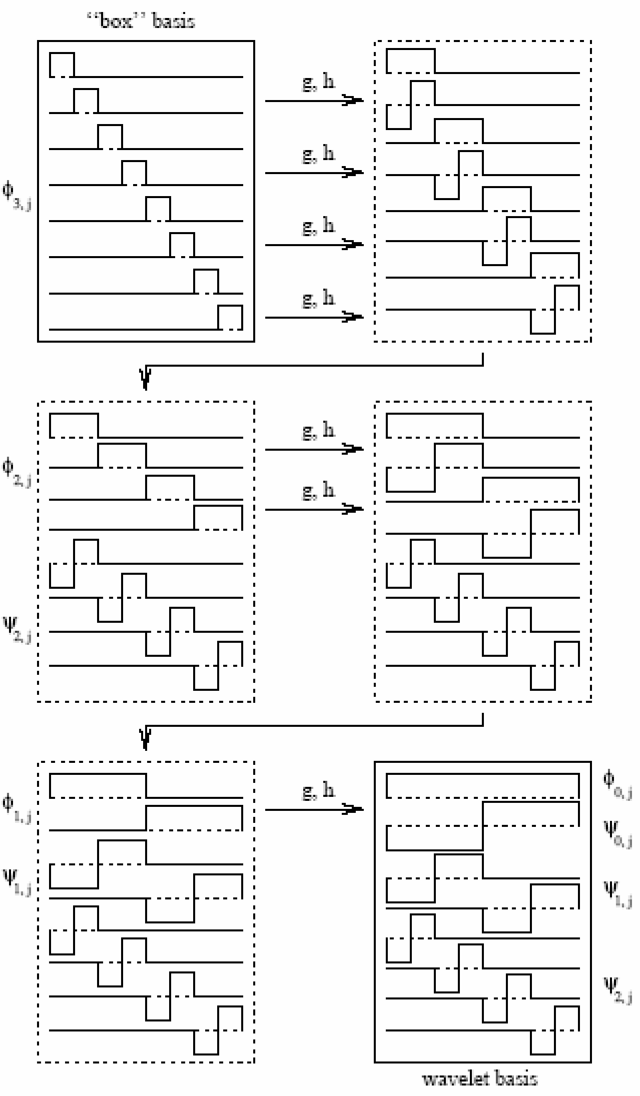
Projections

The projection of the kernel onto some set of basis functions is much like ‘ordinary’ projection in $\mathbf{R}^3$, but instead of projection onto vectors, our ‘axis’ are functions. The projection of a function B(s) onto a set of basis functions $N_i(s)$ can be expressed as a linear combination of the $N_i(s)$:

$B(s) \approx \hat{B}(s) = \sum_{i=1}^{n} B_i N_i(s)$, where B_i are scalar coefficient with respect to the chosen basis.

The projected equation is an approximation of the original integral equation. Depending on what set of basis functions is chosen, the approximation will differ also. Higher order basis functions results in smoother and better approximations than those of lower orders, however, they also require more work.

Wavelets



Wavelets form hierarchical bases that offers alternative bases for finite dimension function spaces (i.e. our N_i:s above).

The simplest wavelet function can be seen in the picture to the right. This is called the *Haar basis*. The upper left corner contains a set of 'box' functions. By combining these boxes pair wise in different ways (adding and subtracting) we may construct another basis for the same function (middle left). We may recursively go on and construct even more bases until we eventually end up at what is called a *wavelet basis* (lower right). Any function that could be expressed as a linear combination of the first set of boxes, can also be expressed by a linear combination of the boxes in the wavelet basis, since those are both bases for the same function space.

Notice, however, than in order to express a linear function in the first basis, we would need to use each and every one of the boxes, whereas in the wavelet basis, only the top function has to be used. Thus a smooth function expressed in the wavelet basis will have most of its coefficients zero or close to zero.

We recall that the kernel function could (for distant patched) be approximated by a low order polynomial. By choosing a appropriate wavelet basis, these low order polynomials, when expressed as a linear combination of the wavelet basis, will have most of its coefficients small or zero. For different wavelet bases this property might apply to constant, linear, quadratic functions and so forth.

In order to find the coefficient for the projection of a function onto the wavelet basis, two methods are available. The first method is by projecting the function onto the original set of functions (e.g. the box function in top left corner) and in turn calculate the coefficients for the function as expressed at the next higher level (middle left), and eventually end up with the coefficient for the wavelet basis. This method, is however not very efficient. A sufficiently smooth function will have many of its coefficients close to zero (so close that they might be neglected and the function still an approximation that suit our needs). The second method exploits this property, and only calculates the coefficients for at lower levels for functions that are not smooth enough. How this is done is, however, one of the subjects for the second summary of this article!