Haar Wavelet : A Solution to Global Illumination With General Surface Properties*

Sumanta N. Pattanaik and Kadi Bouatouch
IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

February 28, 1996

Abstract

This paper presents a method for solving the problem of global illumination for general environments, using projection of the radiance function on a set of orthonormal basis functions. Wavelet scaling functions form this basis set. The highlights of the paper are : it (i) points out the difficulty associated with the straightforward projection of the integral operator associated with the radiance equation and proposes a method for overcoming this difficulty, (ii) gives the data structure and algorithm for illumination solution in environments containing diffuse and non-diffuse reflecting surfaces, and (iii) proposes the use of bi-orthogonal wavelet for the radiance function reconstruction at the time of rendering. Actual implementation has been carried out using the Haar wavelet basis. The main reason for using Haar basis is that it makes the projection of the integral operator, as well as the computation of the inner product of the integral kernel with its basis functions much simpler. However, the algorithm and data structures presented are not restricted to the Haar basis alone.

1 Introduction

Computation of global illumination in an environment requires the solution of linear integral equations. In general, closed form solution does not exist for such equations. So, one resorts to numerical solution methods. Projection method [1] is one such numerical solution method. One comes across the very first explicit use of this method for the global illumination solution in [2]. Much recently [3, 4, 5] there has been a greater surge of interest in application of this method to the illumination problem. These methods defer in their choice and the number of basis functions for carrying out the projection. Of the various choices of

*To appear in 1994 EG Rendering Workshop Proceedings

Figure 1: Three point geometry.

basis functions the wavelets seem to have the edge over others [3, 6] because (i) the hierarchical decomposition and reconstruction characteristics [7] of wavelets allows the use of variable number of basis functions in the projection of functions involved in the same integral equation, (ii) they provide a handle [8] for adaptively deciding on the number of basis functions. So far, the application of wavelets (or for that matter of projection methods) to the illumination problem has been limited to the solution of radiosity equation.

In this paper we have attempted to use wavelet projection method for solution of the general radiance equations. The work presented here may be seen as bringing the brute-force discretisation method of [9] and hierarchical discretisation method of [10] into the framework of functional projection technique. It thus opens up scopes of using higher order basis functions for use in projection-based illumination computation methods.

The organisation of the paper is as follows. We first describe the general radiance equation. Then discuss the functional projection and pose the difficulty involved in projecting the integral operator involved in the equation. We then propose a solution using wavelet basis function, in particular using Haar wavelet basis function. We then provide the data structure and algorithm for the solution of radiance equation using this projection.

2 Three Point Radiance Equation

Radiance from a point $\bar{x}'$ of a surface p towards the point $\bar{x}''$ of surface q, $L_{pq}(\bar{x}', \bar{x}'')$, can be written as [11] :

$$L_{pq}(\bar{x}', \bar{x}'') = \epsilon_{pq}(\bar{x}', \bar{x}'') + \sum_{r} \int_{A_r} \kappa_{rpq}(\bar{x}, \bar{x}', \bar{x}'') L_{rp}(\bar{x}, \bar{x}') d\bar{x} \tag{1}$$

where $\epsilon_{pq}(\bar{x}', \bar{x}'')$ is the radiance due to emission from $\bar{x}'$ along $\bar{x}''$, $\bar{x}$ is a point on the surface r *i.e* $\bar{x} \in A_r$,

$$\kappa_{rpq}(\bar{x}, \bar{x}', \bar{x}'') = \frac{f_p(\bar{x}', \Theta_{\bar{x}\bar{x}'}, \Theta_{\bar{x}'\bar{x}''}) \cos\theta_{\bar{x}\bar{x}'} \cos\theta_{\bar{x}'\bar{x}}}{|\vec{\bar{x}\bar{x}'}|^2} v(\bar{x}, \bar{x}')$$

f is the surface *brdf*, $v(\bar{x}, \bar{x}')$ is the visibility between $\bar{x}$ and $\bar{x}'$.
Solution of equation (1) using projection method will require the expansion of the functions and the integral operator involved in the equation in some basis. Orthonormal wavelet scaling functions $\phi_{J,k}$ with compact support [12]

at some appropriate resolution J can form such basis. For the convenience of explanation, in the following discussions of this section, we shall assume the environment to be 2D (flatland). Also for the uniformity in the variable space, we shall change the radiance equation to a parametric form as follows:

$$L_{pq}(u',u'') = \epsilon_{pq}(u',u'') + \sum_r \int_{u=0}^{1} K_{rpq}(u,u',u'')L_{rp}(u,u')du \tag{2}$$

where $K_{rpq}(u,u',u'') = \kappa_{rpq}(x,x',x'')\frac{dx}{du}$ and the parameters u, u', u'' ranging between 0 and 1 span the flatland surfaces r, p and q respectively. By similar reasoning, in 3D space $K_{rpq}()$ will be a 6 variate function with parameters u, v, u', v', u'', v'' and each u, v pair taking the value in a unit square will span the full area of the surface.
Now the expansion of the radiance and the kernel functions can simply be written as:

$$\begin{aligned} L(u',u'') &= \sum_{k,l} L^J_{k,l}\phi_{J,k}(u')\phi_{J,l}(u'') &(3)\\ K(u,u',u'') &= \sum_{k,l,m} K^J_{k,l,m}\phi_{J,k}(u')\phi_{J,l}(u'')\phi_{J,m}(u) &(4)\end{aligned}$$

where coefficients of expansion $L^J_{k,l}$ and $K^J_{k,l,m}$ are inner product of the functions $L()$ and $K()$ respectively with their corresponding basis functions. *i.e.*

$$\begin{aligned} L^J_{k,l} &= \langle L(u',u''), \phi_{J,k}(u')\phi_{J,l}(u'')\rangle \\ K^J_{k,l,m} &= \langle K(u,u',u''), \phi_{J,k}(u')\phi_{J,l}(u'')\phi_{J,m}(u)\rangle \end{aligned}$$

However, unlike in the case of diffuse radiance equation [3, 6], such straightforward expansion of the integral operator in equation (2) is difficult. This difficulty arises because the integration is over one variable where as the radiance function inside is a function of two variables. Thus the expansion leads to:

$$\begin{aligned} & \int_u K(u,u',u'')L(u,u')du \\ = & \int_u \left[\sum_{k,l,m} K^J_{k,l,m}\phi_{J,k}(u')\phi_{J,l}(u'')\phi_{J,m}(u) \sum_{m',k'} L^J_{m',k'}\phi_{J,k'}(u')\phi_{J,m'}(u) \right] du \\ = & \sum_{k,l,m} K^J_{k,l,m}\phi_{J,k}(u')\phi_{J,l}(u'') \sum_{m',k'} \phi_{J,k'}(u')L^J_{m',k'} \int_u \phi_{J,m}(u)\phi_{J,m'}(u)du \\ = & \sum_{k,l,m} K^J_{k,l,m}\phi_{J,k}(u')\phi_{J,l}(u'') \sum_{k'} \phi_{J,k'}(u')L^J_{m,k'} \end{aligned}$$

The derivation of the last step is due to the orthonormality of the function ϕ, which implies that $\int_u \phi_{J,m}(u)\phi_{J,m'}(u)du = \delta_{m,m'}$. So the summation term over m' disappears. However, unlike the expansion of other functions in the equation (2), the right hand side of the above equation is not a linear combination of only $\phi_{J,k}(u')\phi_{J,l}(u'')$'s.

The compact support of the function ϕ simplifies the above equation a bit further to give:

$$\int_u K(u,u',u'')L(u,u')du = \sum_{k,l,m} K^J_{k,l,m}\phi_{J,l}(u'') \sum_{k'=k-M+1}^{k+M-1} \phi_{J,k}(u')\phi_{J,k'}(u')L^J_{m,k'}$$

where M is the number of vanishing moments of the wavelet function.
Luckily, it is possible to express each of the $\phi_{J,k}(u')\phi_{J,k'}(u')$ terms in the above equation as a linear combination of $\phi_{J,n}(u')$'s [13]. Particularly, with scaling function in the Haar basis, such expansion is simply the following.

$$\sum_{k'=k-M+1}^{k+M-1} \phi_{J,k}(u')\phi_{J,k'}(u') = \phi_{J,k}(u')\phi_{J,k}(u') = 2^{J/2}\phi_{J,k}(u')$$

Here k' takes only one value which is k. This is because in the Haar basis $M = 1$ and $\phi_{J,k}(u')$ has a constant value of $2^{J/2}$ within its support.
Thus, now we can write the expansion of the integral operator as:

$$\int_u K(u,u',u'')L_{rp}(u,u')du \quad = \quad 2^{J/2}\sum_{k,l}\phi_{J,k}(u')\phi_{J,l}(u'')\sum_m K^J_{k,l,m}L^J_{m,k} \quad (5)$$

As the integral term in equation (2) actually represents the reflected radiance, we shall refer to it as $\hat{L}_{rpq}(u',u'')$ and will thus have the following expression.

$$\hat{L}_{rpq}(u',u'') = 2^{J_{rpq}/2}\sum_{k,l}\phi_{J_{rpq},k}(u')\phi_{J_{rpq},l}(u'')\sum_m K^{J_{rpq}}_{k,l,m}L^{J_{rpq}}_{m,k} \quad (6)$$

$\hat{L}_{rpq}()$ carries subscript rpq because it represents reflected radiance from a point u' on surface p towards the point u'' on surface q, due to the incoming radiance from all the points u of surface r. Similarly, J_{rpq} is the appropriate resolution for the expansion of the kernel K_{rpq} using the wavelet scaling function. The method of finding appropriate J_{rpq} will be given in the subsequent section.

Now we can rewrite equation (2) for the geometry in figure 1 as:

$$L_{pq}(u',u'') = \epsilon_{pq}(u',u'') + \hat{L}_{rpq}(u',u'') \quad (7)$$

Using the expansions given in equations (3) and (6) we arrive at the following set of linear equations.

$$L^{J_{rpq}}_{k,l} = \epsilon^{J_{rpq}}_{k,l} + 2^{J_{rpq}/2}\sum_m K^{J_{rpq}}_{k,l,m}L^{J_{rpq}}_{m,k} \quad (8)$$

Allowing for the contribution from more than one surface r, *i.e.*,

$$L_{pq}(u',u'') = \epsilon_{pq}(u',u'') + \sum_r \hat{L}_{rpq}(u',u'') \tag{9}$$

we will have the linear equation set :

$$L_{k,l}^{J_{rpq}} = \epsilon_{k,l}^{J_{rpq}} + \sum_r 2^{J_{rpq}/2} \sum_m K_{k,l,m}^{J_{rpq}} L_{m,k}^{J_{rpq}} \tag{10}$$

Similarly for every pair of surfaces p, q in the environment we will have its corresponding linear set. All these linear equations can be solved together by using an iterative method discussed in the following section.

We have come across similar linear equation formulations for the global illumination involving glossy surfaces in [10]. However, there is a significant difference between the formulations. The linear expression of [10] has been derived by multiplying both sides of equation (2) by a term $G(\bar{x}',\bar{x}'')dx''dx'$ and integrating the resulting equation over the surface p and surface q. *i.e.*:

$$\begin{aligned} \int_{A_p}\int_{A_q} L_{pq}(\bar{x}',\bar{x}'')G(\bar{x}',\bar{x}'')d\bar{x}''d\bar{x}' \quad &= \quad \int_{A_p}\int_{A_q} \epsilon_{pq}(\bar{x}',\bar{x}'')G(\bar{x}',\bar{x}'')d\bar{x}''d\bar{x}' \\ + \sum_r \int_{A_p}\int_{A_q}\int_{A_r} \kappa_{rpq}(\bar{x},\bar{x}',\bar{x}'') \quad L_{rp} \quad (\bar{x},\bar{x}')d\bar{x}G(\bar{x}',\bar{x}'')d\bar{x}''d\bar{x}' \end{aligned} \tag{11}$$

where $G(\bar{x}',\bar{x}'') = \frac{\cos\theta_{x'x}\cos\theta_{x'x''}}{|\bar{x}'\bar{x}''|^2} v(\bar{x}',\bar{x}'')$.

This integration actually leads to an expression of the total flux reaching surface q due to the surface p. If the surfaces p, q and r are very small (*i.e.* the environment has been prediscretised to small patches) and the radiance function over patch p towards the patch q is constant and that over patch r towards the patch p is constant then the both side of the equation (11) can be divided by $\int_{A_p}\int_{A_q} G(\bar{x}',\bar{x}'')d\bar{x}''d\bar{x}'$ to give a simple linear radiance equation. Thus, in [10] the linear equations represent the expressions for the *pre-assumed constant* radiance over a *small patch* towards another *small patch* in an prediscretised environment; whereas in our derivation the linear equations represent the expression for the coefficients of expansion of the radiance function projected over the wavelet basis functions.

3 Environment with Diffuse and Non-diffuse Surfaces

The formulations derived in the previous section assumes general reflection properties for all the surfaces in the environment. As far as the computational complexity is concerned, it may be beneficial to assume that the environment also contains surfaces with diffuse reflection/emission properties. The benefit arises from the directional independence of the radiance from the diffuse surfaces. So for diffuse surfaces the general radiance expression (2) simplifies to:

$$L_p(u') = \epsilon_p(u') \quad + \quad \sum^{nd} \int_{u=0}^{1} K_{rp}(u,u') L_{rp}(u,u') du + \sum^{d} \int_{u=0}^{1} K_{rp}(u,u') L_r(u) du$$

where nd and d are respectively the number of non-diffuse and diffuse surfaces in the environment and p is the diffuse surface,

$$K_{rp}(u,u') = \rho_p(x') \frac{\cos\theta_{x'x} \cos\theta_{xx'}}{\pi \left|\bar{xx'}\right|^2} \frac{dx}{du}$$

and $\rho_p(x')$ is the diffuse reflectivity.
Similar equation can be written for the non-diffuse surface. Projecting such equations on the wavelet basis, we will arrive at the following linear set of equations

$$L_k^{J_{rp}} = \epsilon_k^{J_{rp}} + \sum_r^{nd} 2^{J_{rp}/2} \sum_m K_{k,m}^{J_{rp}} L_{m,k}^{J_{rp}} + \sum_r^{d} \sum_m K_{k,m}^{J_{rp}} L_m^{J_{rp}} \quad \text{for } p \text{ diffuse} \tag{12}$$

and

$$L_{k,l}^{J_{rpq}} = \epsilon_{k,l}^{J_{rpq}} + \sum_r^{nd} 2^{J_{rpq}/2} \sum_m K_{k,l,m}^{J_{rpq}} L_{m,k}^{J_{rpq}} + \sum_r^{d} \sum_m K_{k,l,m}^{J_{rpq}} L_m^{J_{rpq}} \quad \text{for } p \text{ nondiffuse.} \tag{13}$$

We now proceed to give a solution method for the global illumination in an environment, using equations (12) and (13).

4 Solution

The data structure and algorithm given in this section are the extensions to our earlier work for diffuse environment [6].

4.1 Data Structure

We start with the data structure for the surface.

```
typedef struct {
    Geom geometric_information;
    Opt optical_information;
    typedef struct INTERACTION{
      void *kernel;                    /* Pointer to 2Point/3Point struct.*/
      struct INTERACTION *next;
    }Interaction *interaction;
    struct{
      int J_{pq};                      /* Max interaction resolution.*/
      float *[L_{k,l}^{J_{pq}}], *[ε_{k,l}^{J_{pq}}];   /* Projection Coeffs.*/
    }*radiance;
}Surface;
```

$[L_{k,l}^{J_{pq}}]$ and $[\epsilon_{k,l}^{J_{pq}}]$ in the data type $Surface$ are the coefficients of projection of radiance and emittance function respectively and have an hierarchical structure. There are $(J_{pq}+1)$ levels in the hierarchy. Each level $J \in [0, J_{pq}]$ has *up to*[1] 2^{4J} coefficients. The values of $[L_{k,l}^{J_{pq}}]$ are set after each gathering by the surface. For diffuse surfaces only a single structure of $radiance$ is necessary. Whereas for each specular surface there will be n such structures where n is the number of surfaces visible to the specular surface.

```
/*  Structure to hold relevant 2 point interaction informations.*/
typedef struct {
    int J_{pr};           /* Resolution of interaction.*/
    int p, k,l;           /* p :  surface-id, k,l ∈ [1,2^{J_{pr}}] its indices.*/
    int r, i,j;           /* r :  surface-id, i,j ∈ [1,2^{J_{pr}}] its indices.*/
    float K_{i,j,k,l}^{J_{pr}};   /* Coeff of 2 point interaction p ↔ r */
}2Point;                  /*   = ⟨K(), φ_{J,i}()φ_{J,j}()φ_{J,k}()φ_{J,l}()⟩.*/
/*  Structure to hold relevant 3 point interaction informations.*/
typedef struct {
    int J_{rpq};          /* Resolution of interaction.*/
    int k, l;             /* k,l ∈ [1,2^{J_{rpq}}] indices of p.*/
    int q, m,n;           /* q :  receiver, m,n ∈ [1,2^{J_{rpq}}] its indices.*/
    int r, i,j;           /* r :  emitter, i,j ∈ [1,2^{J_{rpq}}] its indices.*/
    float K_{i,j,k,l,m,n}^{J_{rpq}};   /* Coeff of 3 point interaction r → p → q */
}3Point;                  /*  =⟨K(), φ_{J,i}()φ_{J,j}()φ_{J,k}()φ_{J,l}()φ_{J,m}()φ_{J,n}()⟩.*/
```

[1] As we shall see later, though in principle there can be so many coefficients the actual number will depend on the interaction of the 2Point or 3Point kernel.

4.2 Algorithm

wavelet_radiance()

```
begin
 compute_kernel();
 Initialise_radiance_coefficients();
 Compute_radiance_solution();
end.
```

compute_kernel()

```
begin
 for each mutually visible surface pair (p,r) do
  if diffuse(p) or diffuse(r) then
    Compute_2point_Kernel(0,r,1,1,p,1,1);
  if non_diffuse(p)
    Compute_3point_Kernel(0,r,1,1,p,1,1,q,1,1) ∀ q visible to p;
  if non_diffuse(r)
    Compute_3point_Kernel(0,p,1,1,r,1,1,q,1,1) ∀ q visible to r;
  endif;
 endfor; /* for each surface pair */
end;
```

The procedures `Compute_2point_Kernel` and `Compute_3point_Kernel` compute the resolution for projection and compute the projection coefficients of the kernel.

The resolution computation is carried out adaptively starting with $J = 0$ and progressively incrementing the J till the kernel smoothness condition is satisfied [6]. The outline of these two algorithms are as follows:

Compute_2point_Kernel$(J, \mathbf{p}, k, l, \mathbf{r}, i, j)$

begin

  Compute $K^{J+1}_{i',j',k',l'}$ /* *by Numerical quadrature.* */

   $\forall\ i' \in [2i-1, 2i],\ j' \in [2j-1, 2j],\ k' \in [2k-1, 2k]$ and $l' \in [2l-1, 2l]$;

  Compute $\left[{}^{\xi}D^{J}_{i,j,k,l}\right]^{\xi=I,II,\ldots,XV}$ from $K^{J+1}_{i',j',k',l'}$'s [2] ;

          /* *Hierarchical Decomposition*/

  if $\left({}^{\xi}D^{J}_{i,j,k,l} \leq Threshold\right)$ $\forall\xi$ then

   Compute $K^{J}_{i,j,k,l}$ from $K^{J+1}_{i',j',k',l'}$;/* *Hierarchical Decomposition*/

   set J_{pr} = J;

   2point = new(**2Point**); **2point* = $\left\{J_{pr}, \mathbf{p}, (k,l), \mathbf{r}, (i,j), K^{J_{pr}}_{i,j,k,l}\right\}$;

   $(\mathbf{p}).interaction$ = new(struct Interaction);

   $(\mathbf{r}).interaction$ = new(struct Interaction);

---

[2] ${}^{\xi}D^{J}_{i,j,k,l}$ is the inner product of $K()$ with the wavelet function ${}^{\xi}\Psi$. In wavelet basis for each multi-variate scaling function there are $2^m - 1$ number of wavelet basis functions where m is the number of variates.

Figure 2: The data-structure to handle Directional Radiance.

```
  (p).interaction.kernel = (r).interaction.kernel = 2point;
 else Compute_2point_Kernel(J+1, p, k', l', r, i', j')
      ∀ i' ∈ [2i−1, 2i], j' ∈ [2j−1, 2j], k' ∈ [2k−1, 2k], l' ∈ [2l−1, 2l];
 endif;
end;
```

Compute_3point_Kernel$(J, \mathbf{r}, i, j, \mathbf{p}, k, l, \mathbf{q}, m, n)$

```
begin
 Compute K^{J+1}_{i',j',k',l',m',n'} /* By numberical quadrature technique. */
  ∀ i' ∈ [2i−1, 2i], j' ∈ [2j−1, 2j], k' ∈ [2k−1, 2k], l' ∈ [2l−1, 2l],
      m' ∈ [2m−1, 2m], n' ∈ [2n−1, 2n];
 Compute [ ^ξ D^J_{i,j,k,l,m,n} ]^{ξ=I,II,...,CXIII} from K^{J+1}_{i',j',k',l',m',n'}'s ³ ;
      /*Hierarchical Decomposition*/
 if ( ^ξ D^J_{i,j,k,l,m,n} ≤ Threshold ) ∀ξ then
  Compute K^J_{k,l,m,n} from K^{J+1}_{i',j',k',l',m',n'}'s;/*Hierarchical Decomposition*/
  set J_rpq = J;
  3point = new(3Point);
  *3point = { J_rpq, (k,l), q, (m,n), r, (i,j), K^{J_rpq}_{i,j,k,l,m,n} };
  (p).interaction = new(struct Interaction);
  (p).interaction.kernel = 3point;
 else Compute_3point_Kernel(J+1, p, k', l', r, i', j', q, m', n')
      ∀ i' ∈ [2i−1, 2i], j' ∈ [2j−1, 2j], k' ∈ [2k−1, 2k],
      l' ∈ [2l−1, 2l], m' ∈ [2m−1, 2m], n' ∈ [2n−1, 2n];
 endif;
end;
```

In the procedure given above the increment in the resolution of the wavelet basis means the regular subdivision of the surface in the parametric domain. Depending on whether a two-point or a three-point kernel is being handled, 2 or 3 surfaces will be subdivided simultaneously. This seems to be inappropriate when some of the surfaces involved in the interaction are disproportionately small. This problem arises because of the uniform parameterisation of the surfaces. Each of the interacting surfaces is a unit square in the parameter space. However, the actual area may be significantly different from each other. To minimise the problem we use a concept of *virtual resolution*. When refinement is necessary, the resolution of the largest of the interacting surfaces is actually

[3] As discussed earlier, for a function with 6 variables there are $2^6 - 1 = 63$ wavelet basis functions.

incremented, (*i.e.* it is subdivided) whereas for the other two only the virtual resolution is incremented (*i.e.* they are not subdivided. If originally a surface is assumed to span a unit area in the parametric domain, after the increment of the virtual resolution the same surface is assumed to span one a quarter area in the same parametric domain). As the resolution of only one of the surfaces is actually incremented, the computational complexity of the adaptive kernel projection is substantially reduced, but care must be taken while computing the projection. To be precise, both for specular and diffuse interaction kernel only 2^2 operations are carried out at each iteration step instead of the 2^6 in the case of specular and 2^4 in the case of diffuse kernel. The consequence of this change is that the data-structure for handling the directional radiance in $surface$ becomes a bit more complicated. Figure 2 gives the pictorial representation of this modified data structure.

compute_radiance_solution()

```
begin
 repeat
  done = true;
  for each surface p do/*Gather radiance through all interaction links.*/
   if diffuse(p) then
    [T^J_{k,l}] = [(p).ε^J_{k,l}]; /* T : temp struct to gather.*/
    for each interaction of p do
     from (p).interaction get {J_pr, (k,l), r, (i,j), K^{J_pr}_{i,j,k,l}};
     K = K^{J_pr}_{i,j,k,l};
     if diffuse(r) then L_from_r = (r).L^{J_pr}_{i,j}
                   else L_from_r = 2^{J_p} × (r).L^{J_rp}_{i,j,k,l}[p];
    T^{J_pr}_{k,l} += K × L_from_r;
    endfor; /* for each interaction of p */
    2d_resonstruct_decompose(J_p, T);
    if ABS(T^0_{1,1} − (p).L^0_{1,1}) > Threshold then
     done = false;
     [(p).L^J_{k,l}] = [T^J_{k,l}];
    endif;
   else /* non-diffuse surface */
    [T^J_{k,l,m,n}[q]] = [(p).ε^J_{k,l,m,n}[q]]  ∀ q ; /* T : temp struct to gather.*/
    for each interaction of p do
     from (p).interaction get {J_rpq, (k,l), q, (m,n), r, (i,j), K^{J_rpq}_{i,j,k,l,m,n}};
     K = K^{J_rpq}_{i,j,k,l,m,n};
     if diffuse(r) then L_from_r = (r).L^{J_rp}_{m,n}
                   else L_from_r = 2^{J_p} × (r).L^{J_rpq}_{i,j,k,l}[p];
```

```
        T^{J_{rpq}}_{k,l,m,n}[q] += K × L_from_r;
      endfor; /* for each interaction of p */
      3d_reconstruct_decompose(J_pq, T[q]) ∀ q;
      for each q do
        if ABS(T^0_{1,1,1,1}[q] − (p).L^0_{1,1,1,1})[q] > Threshold then
          done = false;
          [(p).L^J_{k,l,m,n}[q]] = [T^J_{k,l,m,n}[q]] ;
        endif;
      endfor; /* for each q */
    endif;
  endfor; /* for each p */
 until (done);
end;
```

Figure 3: Simple scene for demonstration.

5 Results

We have applied the method discussed to simple environments. For the environment shown in figure 3 with an emitter, a reflector and a receiver we have given here two images. Images have been created to show the direction independent radiance distribution over the receiving surface due to the changing reflecting property of the reflector. The specular surface has been modeled as rolled aluminum using Ward's reflection model [14]. Figure 4 is created by associating diffuse reflection property with the reflector. For this the algorithm generated 4616 two point nodes and 205346 two point interactions. Figure 5 is created by associating specular reflection property to the reflector. It required 986 two point nodes, 16301 two point interactions, 420625 three point directional nodes and 335811 three point interactions.

6 Discussions

In spite of the effort of making the projection as accurate as possible, one makes the approximation by choosing a non-zero threshold and by choosing a limit on the maximum resolution J. In our case we had chosen the threshold as 0.0001 and the maximum resolution as 8. In the process, we have been able to capture the diffuse radiance distribution almost accurately when the reflector is diffuse (Figure 4). However, the same is not true when the reflector is specular (Figure 5). So for a better rendering of the computed image, smoothing in some form or other is necessary. Instead of choosing some extraneous smoothing procedure, we propose to carry out the function reconstruction using bi-orthogonal wavelets[15], with which smoothing comes naturally. Using bi-orthogonal wavelets the expansion of a function is as follows:

$$f = \sum < f, \phi_{J,k} > \phi'_{J,k}$$

where each of the basis sets $\{\phi_{J,k}\}$ and $\{\phi'_{J,k}\}$ is not necessarily orthogonal in itself but they are orthogonal to each other. *i.e.*

$$< \phi_{J,k}, \phi'_{J,k'} >= \delta_{kk'}$$

In our application, as one of the basis set is already Haar basis ($\{\phi_{J,k}\}$), all we need is another wavelet basis set which is orthogonal to Haar. At present we are experimenting with spline wavelet basis, which is orthogonal to Haar [16], as the $\{\phi'_{J,k}\}$ for the radiance reconstruction during rendering.

The radiance function computed by the method discussed in this paper is defined from a point on a surface to a point on another surface. So if the final purpose of the computation is rendering, which is most often the case, then it is necessary that the computation be carried out in an enclosure. However, if one is interested in rendering only one or a few number of particular view(s), then the enclosure requirement can be avoided by introducing the eye piece as a hypothetical surface into the environment [11, 10]. This hypothetical surface is capable of receiving illumination, so takes part in the 3 point interactions, but does not obstruct or reflect the illumination.

The algorithm discussed above is still not fast enough for a general environment. We attribute is to the fact that the surface subdivision is entirely dependent on the kernel. Though theoretically it seems to be correct, it may be an overkill in the cases where the real contribution of the interacting surfaces to each other is not very significant. The knowledge of this information may help us in deciding on a coarser subdivision (*i.e.* lower resolution) by appropriate choice of the threshold in the kernel computation. As we do not have the exact information, one better way will be to carry out the kernel determination interactively along with the process. That is, we follow an illumination shooting step instead of a gathering step. As in the usual shooting operation we start with the emitting surface(s) and then proceed to the bright-most surface till the unshot energy is minimum. The kernel computation also precedes in the same fashion, starting along with the emitter and then with the brightmost surface and so on. Adapting the algorithm given above to shooting requires minimal change.

References

[1] L. M. Delves and J. L. Mohamed. *Computational Methods for Integral Equations.* Cambridge University Press, 1985.

[2] Paul Heckbert. *Simulating Global Illumination Using Adaptive Meshing.* PhD thesis, June 1991.

[3] Steven Gortler, Peter Schroder, Michel F. Cohen, and Pat Hanrahan. Wavelet radiosity. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):221–230, 1993.

[4] Harold R. Zatz. Galerkin radiosity. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):213–220, 1993.

[5] Roy Troutman and Nelson L. Max. Radiosity algorithms using higher order finite element methods. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):209–212, 1993.

[6] Sumanta N. Pattanaik and Kadi Bouatouch. Fast wavelet radiosity method. *Computer Graphics Forum, Special Eurographics '94 Conference Proceedings Issue*, Sept. 1994.

Figure 4: Diffuse Reflector.

Figure 5: Specular Reflector.

[7] Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on PAMI*, 11(7):674–693, 1989.

[8] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms i. *Communications on Pure and Applied Mathematics*, XLIV:141–183, 1991.

[9] Dave S. Immel, Michael Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):133–142, Aug. 1986.

[10] Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflections. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 24(4):155–162, August 1993.

[11] Christian Bouville, Kadi Bouatouch, Pierre Tellier, and Xavier Pueyo. A theoretical analysis of global illumination models. *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 53–66, June 1990.

[12] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, XLI:909–996, 1988.

[13] Bernard Delyon and Anatoli Juditsy. *personal communication*, IRISA, 1994.

[14] Gregory J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics (SIGGRAPH '92 Proceedings)*, 26(4):265–272, 1992.

[15] Albert Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavlets. *Communications on Pure and Applied Mathematics*, XLV:485–560, 1992.

[16] Bernard Delyon. Ondelettes orthogonales et biorthogonales. Technical report, IRISA, Rennes, No. 732, November 1993.