

Discontinuity Meshing and Hierarchical MultiWavelet Radiosity

Kadi Bouatouch and S. N. Pattanaik

SIAMES, IRISA, Rennes, France

Abstract

In this paper we propose an extension to the work of Lischinski *et al* [1] on combining hierarchical radiosity method and discontinuity meshing. The extension proposed here supports higher order radiosity function at the estimation step itself. We have carried out this extension using Multi-Wavelet basis functions. Unlike the multi-wavelet radiosity work of Gortler *et al* [2] which uses regular quad-tree subdivision, we carry out the subdivision along the discontinuity boundary. In other words, instead of deriving the finer basis functions by uniform parametric dilations of a mother basis function, we derive the finer basis by mapping the mother basis to arbitrary subdomains created during the subdivision along the discontinuity boundary. To carry out *push* and *pull* operations, which are crucial to the hierarchical algorithm, we compute the necessary filter functions. Our method combines the advantages of wavelet radiosity and discontinuity meshing. The preliminary result shows significant computational improvement.

1 Introduction

As of today, hierarchical method is the state of the art method of solving radiosity in any general environment. This method was first introduced in [3, 4] to estimate radiosity as piecewise constant functions over the environment. Subsequently by using multi-wavelet basis functions with higher vanishing moments (> 1) the method was extended [2] to directly estimate radiosity as piecewise higher order polynomial functions. Directly computing higher order functions implies lesser amount of discretisation and hence lesser overall effort in the illumination computation. These methods require a regular discretisation of surfaces. The surfaces needing subdivisions were subdivided by half in each of their parametric dimensions. This type of subdivision gives the best average performance when it is not at all possible to derive the complexity of the underlying radiosity function. However, in certain cases it may be possible to derive some information on the complexity of the function. In such case, instead of carrying out the regular subdivision to finally focus on the complex region, it will be

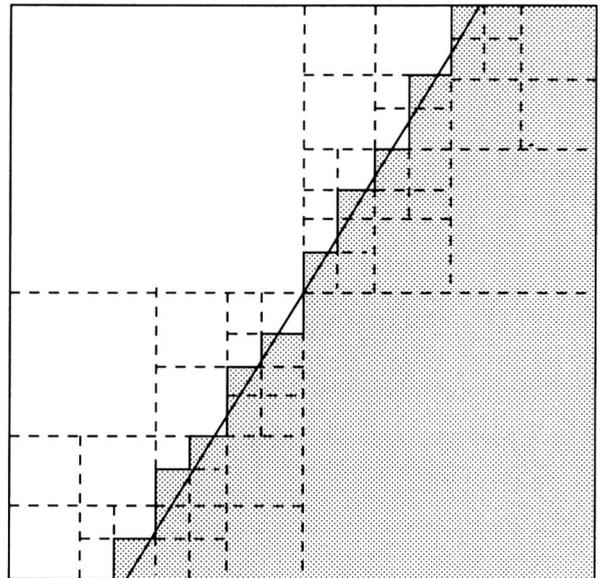


Figure 1: Discontinuity and Wavelet Dilation Functions.

most appropriate to use the derived complexity information to directly localise the complexity. We shall take an example of a shadow discontinuity shown in figure 1. Localising this discontinuity by regular subdivision will require a very large number of subdivisions. However, if the discontinuity can be known by some other extraneous method (say discontinuity meshing) then a single subdivision of the surface will be sufficient to capture the complexity. This example tends to suggest that at the preprocessing stage we carry out the subdivision of the environment using a discontinuity mesher. However, such preprocessing approach are extremely expensive because,

- in the absence of the knowledge of the actual illumination distribution in the environment, the prediscritisation step is likely to create far more number of discrete surfaces than actually required, and



- prediscretisation would mean an increase in the number of individual interacting elements in the environment, hence a quadratic increase in the illumination computation time.

To overcome this problem Lischinski *et al* [1] introduced the discontinuity driven hierarchical radiosity method in which they combined the advantages of hierarchical radiosity and discontinuity meshing. They carried out the normal hierarchical radiosity algorithm to compute interaction between surfaces in the environment, but at the subdivision step of the hierarchical algorithm, instead of using the regular quad-tree subdivision, they used irregular subdivision of surfaces along discontinuity boundaries. However, they restricted their method to the estimation of piecewise constant radiosity functions. For better visual reconstruction of the estimated radiosity function, only in a post-processing step they carried out higher order (quadratic) interpolation.

We extend the work of Lischinski *et al* by directly estimating piecewise higher order radiosity function. This work combines the advantages of wavelet-radiosity and discontinuity meshing. As in Gortler *et al* [2] we have used Multi-wavelet basis function to carry out the extension.

The organisation of the paper is as follows. We briefly introduce the function approximation and hierarchical radiosity. Then we derive the filters to carry out the crucial push/pull operation. Finally we demonstrate its successful application to a simple test environment.

2 Radiosity Function Approximation and Hierarchical Algorithm

In an environment with diffusely reflecting and/or emitting surfaces the radiosity of any surface p can be expressed using the following equation:

$$B_p(\bar{x}) = E_p(\bar{x}) + \sum_{q=1}^N \int_{A_q} \kappa_{q \rightarrow p}(\bar{x}, \bar{y}) B_q(\bar{y}) d\bar{y} \quad (1)$$

where N is the number of surfaces in the environment, \bar{x} and \bar{y} are points and $B_p(\bar{x})$, $B_q(\bar{y})$ the radiosity functions over the surfaces p and q respectively, $E_p(\bar{x})$ is the emittance function over the surface p and $\kappa_{q \rightarrow p}$, the kernel of the integral operator, represents the interaction between surfaces q and p and can be expressed as

$$\kappa_{q \rightarrow p}(\bar{x}, \bar{y}) = \rho_p(\bar{x}) \frac{\cos \theta_x \cos \theta_y}{\pi r^2} V(\bar{x}, \bar{y}) \quad (2)$$

where ρ is the diffuse reflectivity at a point, r the distance between \bar{x} , \bar{y} , V the visibility between \bar{x} , \bar{y} , and θ_x , θ_y respectively are the angles between the line joining \bar{x} and \bar{y} , and the surface normals at those points.

Solution of this type of equation is often carried out by using function approximation technique. This method seeks for the exact solution by projecting the involved functions onto a finite dimensional space, *i.e.* the radiosity function B is approximated by a linear combination

of basis functions:

$$B(\bar{x}) = B(u, v) \approx \sum_{k=1}^n b_k N_k(u, v)$$

where n is the number of basis functions, b_k 's are unknown approximation coefficients and N_k 's are the chosen basis functions.

A set of linear equations [5] of the type

$$b_{p,i} = e_{p,i} + \sum_{q=1}^N \sum_{j=1}^{n_q} K_{q \rightarrow p, ij} b_{q,j} \quad (3)$$

where $i = 1..n_p$ and n_p , n_q are the number of approximation coefficients for $B_p(u, v)$ and $B_q(s, t)$, and

$$\begin{aligned} b_{p,i} &= \int_{u=0}^1 \int_{v=0}^1 N_i(u, v) B_p(u, v) du dv, \\ e_{p,i} &= \int_{u=0}^1 \int_{v=0}^1 N_i(u, v) E_p(u, v) du dv, \\ b_{q,j} &= \int_{s=0}^1 \int_{t=0}^1 N_j(s, t) B_q(s, t) ds dt, \\ K_{q \rightarrow p, ij} &= \int_{u=0}^1 \int_{v=0}^1 \int_{s=0}^1 \int_{t=0}^1 \kappa_{q \rightarrow p}(u, v, s, t) \\ &\quad N_i(u, v) N_j(s, t) \left\| \frac{\delta \bar{y}}{\delta s} \times \frac{\delta \bar{y}}{\delta t} \right\| ds dt du dv. \end{aligned}$$

are then derived from equation 1 by using orthogonal basis functions and Galerkin error minimisation technique. The solution is carried out by following an iterative method such as Gauss-Seidel method or Southwell relaxation method till convergence. The complexity of solving this system for the whole environment is $O(\sum_p n_p)^2$. The accuracy of the resulting solution depends on how correctly the radiosity function on each surface has been approximated to arrive at the linear equation. The brute force method of improving the accuracy is to increase the number of basis function n_p for the approximation of radiosity function of each surface p . However, as the complexity of the technique indicates the quadratic increase in computation time with the increase in the number of basis function, one must make optimal use of the number of basis function for any approximation.

Hierarchical algorithm [4] addresses this problem by making some crucial observations. They are :

- To write an expression for the unknown radiosity function of surface p of the type given in equation (3), we need the expression of radiosity function of all other surfaces q which are fully or partially visible to p .
- It is possible to decide on the maximum discretisation of each surface for any given accuracy in the solution of the radiosity function.
- If a particular accuracy in computation of radiosity of surface p requires a finer discretisation of surface



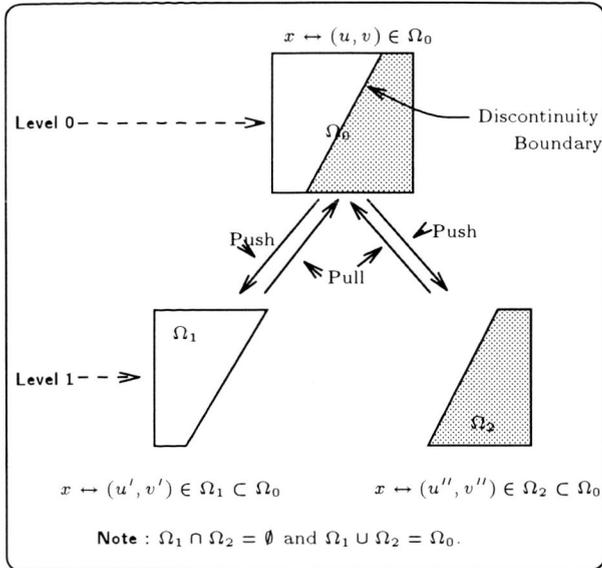


Figure 2: Push-Pull operation.

q , then it is not necessary that for the same computational accuracy in radiosity of another surface r will require the discretisation of q to the same fineness. Probably a coarser discretisation or much finer discretisation may be required.

The last observation is very crucial because if we can use different levels of discretisation of a particular surface to set up the system of equations of the type (3) then the overall complexity of the solution is bound to be much less than $\mathcal{O}(\sum_p n_p)^2$ where n_p is the maximum discretisation of the surface p . Making use of this observation demands that one must

- associate a hierarchy of subdivisions with each surface and
- at any time during the iterative solution of radiosity, maintain with each level of the hierarchy the radiosity information commensurate to the level.

Maintaining radiosity function at different levels during the iteration requires 2 operations known as *push* and *pull* operations. In the next section we describe the basic ingredients necessary for these operations.

3 Push/Pull Operation

Let us consider the simplest case of an one-level surface discretisation shown in figure 2. Unlike the quad-tree uniform subdivision, here the subdivision is binary and non-uniform. Given the approximation of a function, $B(\bar{x})$, defined over the level 0, the *push* operation computes the approximation of the the function at level 1. And similarly, given the approximation at level 1, the *pull* operation computes the approximation of the function at level 0. Let us define three sets of orthonormal

basis functions $N_m^{(0)}(\bar{x})$, $N_m^{(1)}(\bar{x})$ and $N_m^{(2)}(\bar{x})$ such that they are zero outside the domains Ω_0 , Ω_1 and Ω_2 respectively, and if we assume that a unit square in parametric domains (u, v) , (u', v') and (u'', v'') span Ω_0 , Ω_1 and Ω_2 respectively, then

$$\begin{aligned} N_m^{(0)}(u, v) &= \mathcal{M}_m(u, v) \\ N_m^{(1)}(u', v') &= \mathcal{M}_m(u', v') \\ N_m^{(2)}(u'', v'') &= \mathcal{M}_m(u'', v'') \end{aligned}$$

where \mathcal{M}_m 's are the scaling function of a multiwavelet basis of vanishing moment M .

With these definitions we can write the approximation of the radiosity function at the given two levels as follows:

Approximation at Level 0 :

$$B(u, v) \approx \sum_{m=1}^{M \times M} b_m^{(0)} N_m^{(0)}(u, v) \quad (4)$$

where

$$b_m^{(0)} = \int_{u=0}^1 \int_{v=0}^1 B(u, v) N_m^{(0)}(u, v) du dv$$

Approximation at Level 1 :

$$\begin{aligned} B(u, v) \approx & \sum_{m=1}^{M \times M} b_m^{(1)} N_m^{(1)}(u'(u, v), v'(u, v)) \\ & + \sum_{m=1}^{M \times M} b_m^{(2)} N_m^{(2)}(u''(u, v), v''(u, v)) \end{aligned} \quad (5)$$

where

$$\begin{aligned} b_m^{(1)} &= \int_{u'=0}^1 \int_{v'=0}^1 B(u(u', v'), v(u', v')) \\ & \quad N_m^{(1)}(u', v') du' dv', \\ b_m^{(2)} &= \int_{u''=0}^1 \int_{v''=0}^1 B(u(u'', v''), v(u'', v'')) \\ & \quad N_m^{(2)}(u'', v'') du'' dv''. \end{aligned}$$

Push Filters : As explained above, *pushing* amounts to computing the approximation coefficients $b_m^{(1)}$'s and $b_m^{(2)}$'s from the approximation coefficients $b_m^{(0)}$'s. We derive them as follows.

$$\begin{aligned} b_m^{(1)} &= \int_{u'=0}^1 \int_{v'=0}^1 B(u(u', v'), v(u', v')) \\ & \quad N_m^{(1)}(u', v') du' dv' \\ / * \text{ Using approx of } B \text{ from eq.4 we get, } * / \\ &= \int_{u'=0}^1 \int_{v'=0}^1 \sum_{n=1}^{M \times M} b_n^{(0)} N_n^{(0)}(u(u', v'), v(u', v')) \end{aligned}$$



$$\begin{aligned}
& N_m^{(1)}(u', v') du' dv' \\
= & \sum_{n=1}^{M \times M} b_n^{(0)} \int_{u'=0}^1 \int_{v'=0}^1 N_n^{(0)}(u(u', v'), v(u', v')) \\
& N_m^{(1)}(u', v') du' dv' \\
= & \sum_{n=1}^{M \times M} c_{m,n}^{(0 \rightarrow 1)} b_n^{(0)} \quad (6)
\end{aligned}$$

where

$$c_{m,n}^{(0 \rightarrow 1)} = \int_{u'=0}^1 \int_{v'=0}^1 N_n^{(0)}(u(u', v'), v(u', v')) N_m^{(1)}(u', v') du' dv'$$

and similarly

$$b_m^{(2)} = \sum_{n=1}^{M \times M} c_{m,n}^{(0 \rightarrow 2)} b_n^{(0)}$$

where

$$c_{m,n}^{(0 \rightarrow 2)} = \int_{u''=0}^1 \int_{v''=0}^1 N_n^{(0)}(u(u'', v''), v(u'', v'')) N_m^{(2)}(u'', v'') du'' dv''$$

The coefficients $c_{m,n}^{(0 \rightarrow 1)}$'s and $c_{m,n}^{(0 \rightarrow 2)}$'s make the push filter.

Pull Filters : These filters are responsible for computing $b_m^{(0)}$'s from $b_m^{(1)}$'s and $b_m^{(2)}$'s. Similar to the derivation of push filters we derive them as follows.

$$\begin{aligned}
b_m^{(0)} &= \int_{u=0}^1 \int_{v=0}^1 B(u, v) N_m^{(0)}(u, v) dudv \\
& / * \text{Using approx of } B \text{ from eq.5 we get, } * / \\
&= \int_{u=0}^1 \int_{v=0}^1 \sum_{n=1}^{M \times M} b_n^{(1)} N_n^{(1)}(u'(u, v), v'(u, v)) \\
& N_m^{(0)}(u, v) dudv \\
&+ \int_{u=0}^1 \int_{v=0}^1 \sum_{n=1}^{M \times M} b_n^{(2)} N_n^{(2)}(u''(u, v), v''(u, v)) \\
& N_m^{(0)}(u, v) dudv \\
&= \sum_{n=1}^{M \times M} b_n^{(1)} \int_{u=0}^1 \int_{v=0}^1 N_n^{(1)}(u'(u, v), v'(u, v)) \\
& N_m^{(0)}(u, v) dudv \\
&+ \sum_{n=1}^{M \times M} b_n^{(2)} \int_{u=0}^1 \int_{v=0}^1 N_n^{(2)}(u''(u, v), v''(u, v)) \\
& N_m^{(0)}(u, v) dudv \\
&= \sum_{n=1}^{M \times M} [c_{m,n}^{(1 \rightarrow 0)} b_n^{(1)} + c_{m,n}^{(2 \rightarrow 0)} b_n^{(2)}]
\end{aligned}$$

where

$$\begin{aligned}
c_{m,n}^{(1 \rightarrow 0)} &= \int_{u=0}^1 \int_{v=0}^1 N_n^{(1)}(u'(u, v), v'(u, v)) \\
& N_m^{(0)}(u, v) dudv, \\
c_{m,n}^{(2 \rightarrow 0)} &= \int_{u=0}^1 \int_{v=0}^1 N_n^{(2)}(u''(u, v), v''(u, v)) \\
& N_m^{(0)}(u, v) dudv.
\end{aligned}$$

The coefficients $c_{m,n}^{(1 \rightarrow 0)}$'s and $c_{m,n}^{(2 \rightarrow 0)}$'s make the push filter.

Relation Between Push and Pull Filters : As we are using orthogonal basis functions, it may be worthwhile to see if there exists any relationship between the *push/pull* filters, which can reduce the computational effort for evaluating these filter coefficients. We shall first try to find a relationship between $c_{m,n}^{(1 \rightarrow 0)}$ and $c_{m,n}^{(0 \rightarrow 1)}$.

$$\begin{aligned}
c_{m,n}^{(1 \rightarrow 0)} &= \int_{u=0}^1 \int_{v=0}^1 N_n^{(1)}(u'(u, v), v'(u, v)) \\
& N_m^{(0)}(u, v) dudv \\
&= \int_{u'=0}^{U'} \int_{v'=0}^{V'} N_n^{(1)}(u', v') \\
& N_m^{(0)}(u(u', v'), v(u', v')) \mathcal{A}(u', v') du' dv'
\end{aligned}$$

where the point (0,0) in the (u, v) domain maps to (0,0) in the (u', v') domain and (1,1) maps to (U', V') and as $\Omega_1 \subset \Omega_0$, both U', V' are greater than 1. $\mathcal{A}(u', v')$ gives the quantitative area relationship between a differential area in (u, v) domain and a differential area in (u', v') domain.

From the definition, $N_n^{(1)}()$ is zero outside the parametric unit square. So we can reduce the integration limit in the above equation to arrive at:

$$c_{m,n}^{(1 \rightarrow 0)} = \int_{u'=0}^1 \int_{v'=0}^1 N_n^{(1)}(u', v') N_m^{(0)}(u(u', v'), v(u', v')) \mathcal{A}(u', v') du' dv' \quad (7)$$

From the equation 7, we see that only when $\mathcal{A}(u', v')$ is a constant function (\mathcal{C}), we can find a simple relationship between the *push* and *pull* filters as follows:

$$\begin{aligned}
c_{m,n}^{(1 \rightarrow 0)} &= \mathcal{C} \int_{u'=0}^1 \int_{v'=0}^1 N_n^{(1)}(u', v') \\
& N_m^{(0)}(u(u', v'), v(u', v')) du' dv' \\
&= \mathcal{C} c_{n,m}^{(0 \rightarrow 1)}
\end{aligned}$$

For example, when we have subdivision by uniform dilation of a factor 2, as in [4, 2] then we have $\mathcal{A}(u', v')$ is constant and is equal to 0.25. So $c_{m,n}^{(1 \rightarrow 0)} = 0.25 c_{n,m}^{(0 \rightarrow 1)}$. However, for arbitrary subdivisions, it will not be possible to find any such simple relationship. Thus one has to compute the push filters and the pull filters separately.

So far we have discussed the operation from the top most level to the next level. The same discussion can



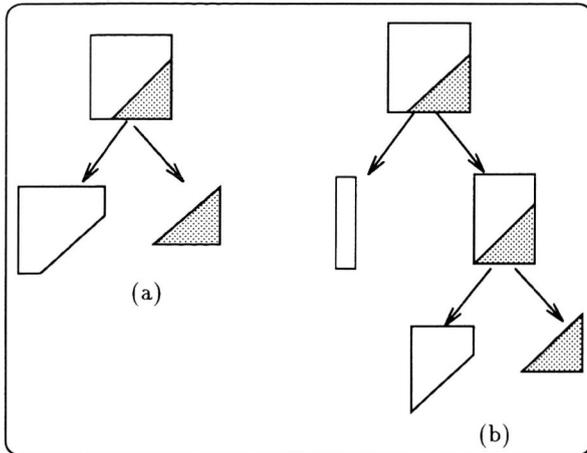


Figure 3: Re-parametrisation Issue.

be extended to any pair of levels. For every level of discretisation we have to compute the equivalent filter function. It must be noted here that similar filters were also required for the *push/pull* using multi-wavelets with uniform dilation. But this uniform dilation allowed one

- to use predefined filter coefficients, and
- the filter coefficients were same between any pair of consecutive discretisation levels.

Whereas, in the push/pull operation with arbitrary discretisation

- the filter function is likely to vary for every pair of consecutive discretisation level, and
- these must be computed at each level during the discretisation process.

4 Re-parameterisation

Subdivision of a biparametric surface along an arbitrary boundary may lead to patches which are difficult to directly parameterise (example: figure 3(a)). All our above discussion assumes that we are able to map the mother multi-wavelet basis function to the domain of the subdivided patch, we must find a mechanism of reparameterisation. To do this, we use a very simple approach. The approach is demonstrated in figure 3(b) which avoids the parameterisation problem posed in figure 3(a). We make sure that each discretisation step leads to discretisation of only one parametric dimension. If it is not so, we introduce another extra step to guarantee this discretisation.

5 Results

We show here the advantages of applying higher order approximation with subdivision across discontinuity boundary using a simple test scene given in figure 4.

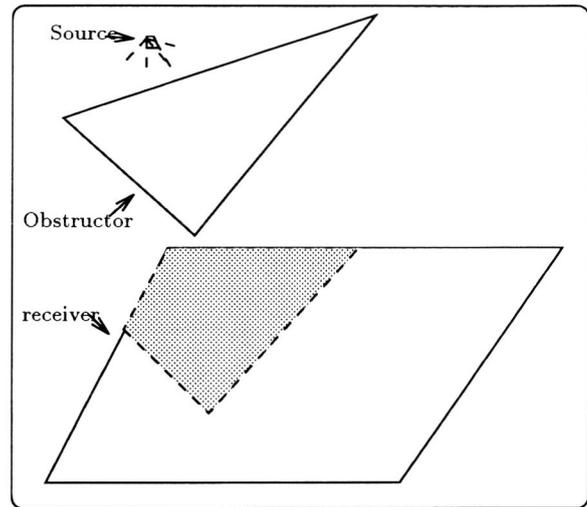


Figure 4: A typical scene.

Figures 5, 6, 7 and 8 compile the results obtained with various strategies. The strategies are respectively:

- uniform subdivision with constant basis function (1776 patches, with subdivision limited to maximum 6 levels of quad_tree),
- subdivision at discontinuity boundary with constant basis function (1019 patches),
- uniform subdivision with multi-wavelet basis function with 4 vanishing moments (262 patches), and
- subdivision at discontinuity boundary with multi-wavelet basis function with 4 vanishing moments (3 patches).

In each of the cases the total number of patches resulting in the process has been given to show the performance improvement. The decision, of whether to subdivide or not, has been taken by consulting an *oracle* similar to the one used in [2] *i.e.* by trying to find if it is possible to get a polynomial fit, of the necessary degree (which in our case is between 0 to 3) within a predefined threshold, for the integration kernel of equation 1. The threshold has been kept same for all the above experiments.

Note that working with multi-wavelet basis function (with $M=4$) involves up to 16 times more effort for each patch in the *given case*, as compared to effort using constant basis function at the same level. So if we compare uniform subdivision results (*i.e.* (a) and (c)), even though higher order basis functions results in a smaller number patches, still then the overall effort far exceeds that due to constant basis function. However, the result due to arbitrary subdivision with higher order basis function performs much favourably compared to all the strategies.

It is a coincidence that in the example given the patches resulting after discontinuity meshing did not require any further subdivision. However, in general further subdivisions may be required. In such case, one can go back to the normal quad-tree subdivision and use the predefined push/pull filters. It is simply a matter of keeping a flag in the hierarchy, which indicates whether the level below is due to uniform meshing or



due to discontinuity meshing.

6 Conclusion

We believe that our proposed technique can be used in the complex environments with substantial benefits. We are carrying out these tests. In the discussion of the paper, we have emphasised on the subdivision along the discontinuity boundary. However, the method is not limited to this. As long as one is able to decide on a best boundary of subdivision one can apply the above method. We are planning to extend the method to the adaptive mesh generation work of Campbell and Fussell [6].

References

- [1] Dani Lischinski, Filippo Tampieri, and Don Greenberg. Combining hierarchical radiosity and discontinuity meshing. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):199–208, Aug. 1993.
- [2] Steven Gortler, Peter Schroder, Michel F. Cohen, and Pat Hanrahan. Wavelet radiosity. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):221–230, 1993.
- [3] Pat Hanrahan and David Salzman. A rapid hierarchical radiosity algorithm for unoccluded environments. *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, 151–171, June 1990.
- [4] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):197–206, July 1991.
- [5] Harold R. Zatz. Galerkin radiosity. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):213–220, 1993.
- [6] A.T. Campbell, III, and Donald S. Fussell. Adaptive mesh generation for global diffuse illumination. *Computer Graphics (SIGGRAPH '90 Proceedings)*, 24(4):155–164, Aug. 1990.



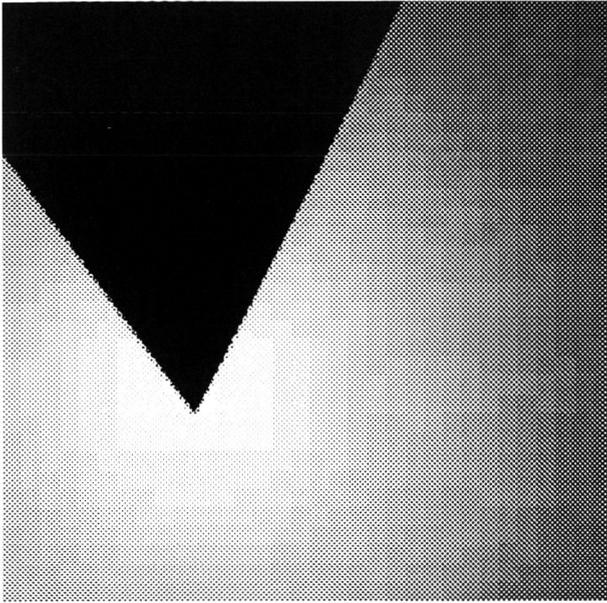


Figure 5: Uniform meshing + Constant Basis.

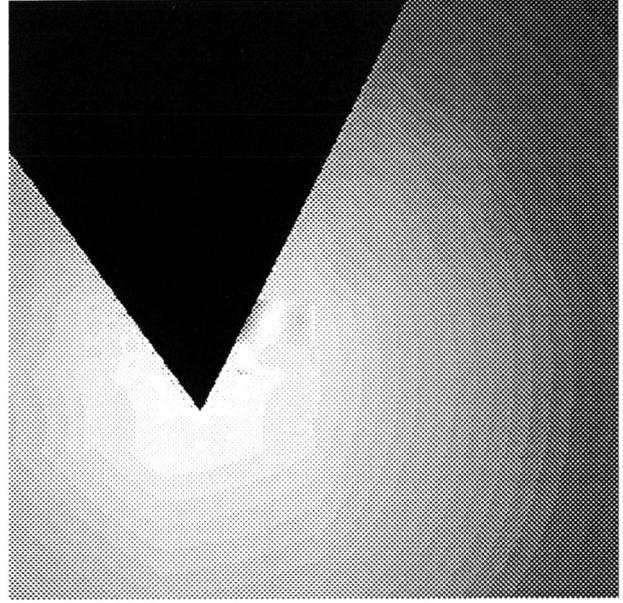


Figure 7: Uniform meshing + Higher Order Basis.

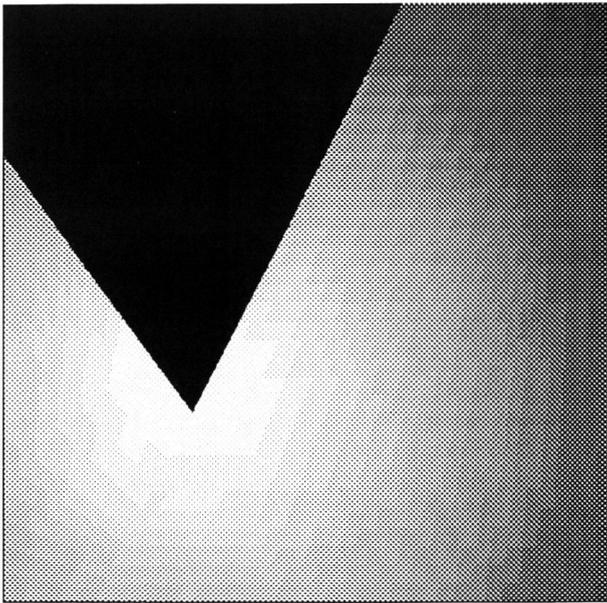


Figure 6: Discontinuity meshing + Constant Basis.

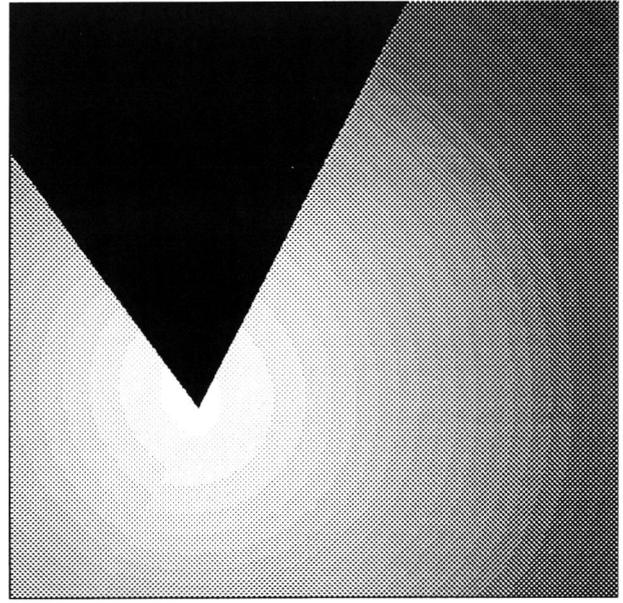


Figure 8: Discontinuity meshing + Higher Order Basis.

