COMBINING HIERARCHICAL RADIOSITY AND LODS

Reynald DUMONT†‡, Kadi BOUATOUCH‡

‡*IRISA, Campus de Beaulieu, 35042 Rennes Cedex, FRANCE*
†*PCG, Rhodes Hall, Cornell University, Ithaca, 14850, NY, USA*

E-Mail: reynald@graphics.cornell.edu, kadi@irisa.fr

Abstract

Clustering and partitioning based methods have drastically improved Hierarchical Radiosity (HR). The first method reduces the number of links but lacks precision since it does not account precisely for the orientations of the patches within the emitting cluster relative to the location of the receiving one. As for the second approach, it is efficient since it partitions large environments into 3D cells and employs ordering strategies that need only a subset of cells for computing illumination at each step of the resolution process. Nevertheless, the computational complexity still remains high when the cells contain many objects of highly detailed geometry. The new HR algorithm, described in this paper, brings solutions to these problems by making use of Levels of Detail (LOD) that approximate the geometry of the object surfaces in the scene.

Keywords: Rendering Algorithm, Global Illumination, Hierarchical Radiosity, LODs.

1 Motivation

Hierarchical radiosity (HR) [1], is widely used for a large range of applications such as: lighting simulation, thermal engineering, radiative transfer within canopies, acoustics, etc. Even for scenes of moderate complexity, HR still is a demanding process in terms of memory and computing resources. This is due to the need for meshing surfaces into elements (or patches), creating links and computing visibility relationships between these elements. For complex building interiors, solutions have been proposed to cope with the memory and computational problems. These solutions rely on a preprocessing step [2, 3, 4] consisting in partitioning the scene into 3D cells and computing a visibility graph that expresses visibility relationships between these cells. Only a subset of cells, concerned with the illumination computation is maintained in memory at a given iteration of the resolution process. Once this preprocessing has been done, ordering strategies are employed for radiosity computation (as in [5]). Even though their efficiency has been proved, these strategies do not reduce the complexity of scenes where cells are composed of a large number of small input polygons that finely describe surfaces or objects of highly detailed geometry. Indeed, in a cell, if the number of polygons is high then so do the number of patches, the number of links as well as the memory needed.

As the flux of a patch is proportional to its surface area, when a small patch is selected in isolation as an emitting patch, its flux (to be emitted) may be insignificant. Consequently, its effect on the environmental illumination may be unnoticeable. On the other hand, if a set of small neighboring patches are selected as emitting patches, their total flux may contribute significantly to the scene illumination, which will speed up the convergence of the resolution process. One way of carrying out this idea is to make use of levels of detail (LOD). More clearly, the surface of an object may be represented by different LOD approximations organized as a hierarchy, called from now on *LOD hierarchy*. The highest level corresponds to the more detailed geometry of the surface. In this way two distant surfaces may interact (in the sense of hierarchical radiosity) at coarse LODs while close surfaces at finer LODs (see figure 1). This allows the number of links to be drastically reduced. As explained later on, accounting for LODs in HR raises some problems that have to be solved: Energy conservation, occlusion, internal reflection, energy distribution similar to push-pull operations.

This paper is organized as follows. Our approach to LODs is outlined in section 2. Then the different steps of our algorithm are described. These steps are: internal emission (section 3.1), energy emission (section 3.2), energy contribution (section 3.3) and finally energy distribution (section 3.4). Before concluding, some results are given in section 4 and 5.

2 Outline

2.1 Problem statement

In classical HR, only facets (the input polygons of the initial geometry) are used and refined during the resolution process. A link associated with a pair of mutually visible facets is called initial link. Let N_{IL} be the number of initial links. The number of links created by a HR algorithm has a lower bound equal to N_{IL}. When the scene contains objects of highly detailed geometry, N_{IL} may be high and so does the final number of links.

To overcome this difficulty, one solution is to make use of HR with clustering techniques [6, 7, 8] or geometric simplification methods [9, 10]. However, these techniques do not account precisely for the orientations of the facets within the emitting cluster relative to the location of the receiving one. In [11], Gibson and Hubbold introduced some improvements to account for surface orientation but their method still makes assumptions. For instance, they make use of the clustering method proposed by Sillion [7] which assumes a cluster as an isotropic participating medium (which is not generally the case) characterized by an extinction factor (difficult to estimate accurately) which often provides block-effect shadow artifacts.

Visibility between clusters has been improved in [12]. In addition, clustering techniques have also been applied to environments containing specular surfaces [13, 14, 15]. These techniques assume that clusters are approximated as point light sources since they rely on intensity. Moreover, details are not always provided about the way the authors account for the orientation of the facets within each cluster (for both energy emission and energy distribution).

As explained in [16], given the scene configuration and the nature of the objects, clustering techniques may result in image artifacts or unexpected behavior. Hasenfratz observed in his paper that for some scenes, lighting simulation could be difficult even with clustering methods. For scenes containing small objects represented by a large collection of polygons existing algorithms are less well-behaved. In presence of such objects, LOD representations seem a natural approach and provide good results.

Our approach is to combine HR and LODs for reducing N_{IL} significantly, which entails a smaller number of links, incurs a lower visibility calculation cost and requires less memory for storing the different data structures involved. Indeed, two surfaces may interact at a coarse LOD if they are sufficiently distant (see figure 1). In this case, these are polygons belonging to these coarser representations that interact mutually instead of facets, patches or surface elements. Such polygons are called *macrofacets*. Our method accounts precisely for the orientations and locations

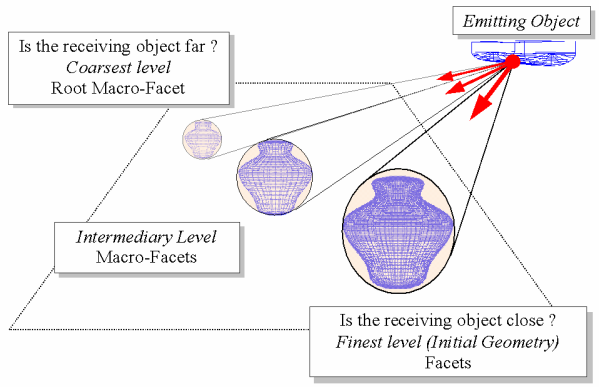

Figure 1: Use of LODs during energetic transfer

of the emitting and receiving macrofacets (and facets) whenever a macrofacet shoots its energy or distributes it to its descendants.

2.2 Definitions

We suppose all the objects within the scene are modeled with polygonal surfaces, each one being represented by a hierarchy of LOD approximations. The root of a hierarchy corresponds to the coarsest polygonal approximation of the surface while a leaf corresponds to the finest one (see figure 2). A non-terminal node and a leaf node are polygons that are called *macrofacet* and *facet* respectively. In an LOD hierarchy (called from now on HLOD), the children of a non-terminal node (macrofacet) may be macrofacets and/or facets not coplanar. Note that an input polygon without LOD approximations is also represented by a LOD hierarchy but composed of only one node. A facet may be subdivided into another hierarchy of patches (HP) by the refinement process employed in any classical HR. In this case, an internal node and a leaf node of a HP are referred to as *patch* and *surface element* respectively.

2.3 Creating LOD hierarchies

The LOD hierarchy associated with a surface can be created with the help of any surface simplification method ([17, 18, 19, 20, 21, 22]). There are broadly three simplification approaches: vertex decimation, edge contraction and surface refinement. The decimation approach removes recursively one or several vertices from the exact geometry while the contraction approach removes edges and replace them with new vertices. Conversely, the refinement approach starts with a coarse approximation of the surface then refines it progressively by inserting new vertices.

In our system, the LOD hierarchies are created during a modeling preprocess. To make this easier, we have developed an interactive software tool using the MOTIF and OpenGL libraries. With a mouse, the user selects a surface and starts the simplification

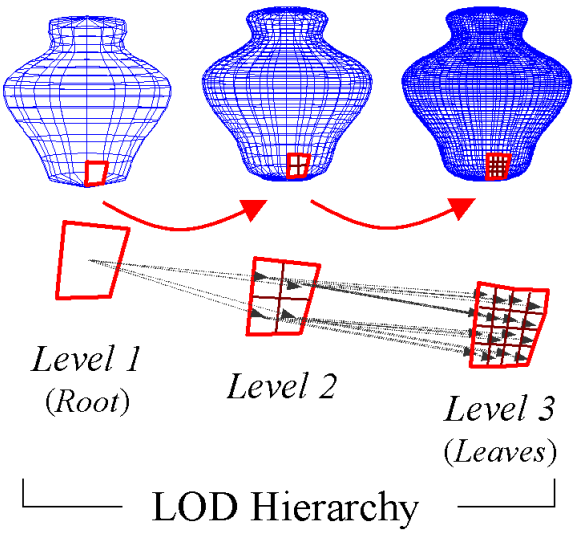

Figure 2: LOD Hierarchy

process giving coarser and coarser approximations (a bottom-up process as in vertex decimation).

2.4 Algorithm overview

In this section we describe only the basic ideas of the algorithm, details are given in the following sections.

Our HR uses a *shooting* technique and removes all the links created during each iteration to save memory. This dynamic link-management is similar to the ones used in ([23] and [24]).

Let us call *root macrofacet* a macrofacet at the root of a LOD hierarchy. The algorithm starts by selecting the emitting root macrofacet E of maximum flux. E has to shoot its flux to each other visible root macrofacet R.

In fact, E and R may not interact directly but through nodes of their respective LOD hierarchies. More precisely, the two LOD hierarchies are traversed top-down and two form factors F_{E-R} and F_{R-E} are estimated for the two current nodes N_E and N_R, belonging to E and R respectively.

F_{E-R} is a point-surface form factor whose apex is the center of the emitting macrofacet (or facet) E and subtended by the receiving macrofacet (or facet) R. F_{R-E} is a point-surface form factor whose apex is the center of the receiving macrofacet (or facet) R and subtended by the emitting macrofacet (or facet) E. When trying to link two primitives, these two form factors do not account for the visibility factor so as to not slow down our HR algorithm. On the other hand, this factor is taken into account once the linking has been performed (note that object representation with LODs offers the possibility to speed-up the visibility determination by using bouding volumes for each object in the scene).

If F_{E-R} and F_{R-E} are below a threshold T_{LOD} then these two nodes are linked and N_E shoots its energy to N_R using F_{E-R} weighted by a visibility factor.

Before shooting the flux of N_E, all the descendants (finer LODs) of N_E exchange one to another their flux according to one iteration of the Jacobi resolution method. This operation will be referred to as *internal shooting* (as explained in section 3.1).

Then, for each N_R a new radiosity B_c of N_E is computed (depending on the location of N_R in the scene) as shown in figure 3 (step 1). This radiosity accounts for the internal exchanges within N_E. From now on, B_c will be called *contextual radiosity* since it depends on the context of N_R. Then N_E shoots its energy towards N_R (figure 3, step 2). Finally, N_R distributes the gathered energy to its descendants (figure 3, step 3).

Note that if the traversal of R's LOD hierarchy leads to a leaf P while F_{E-R} still remains higher than T_{LOD} then P is subdivided into a hierarchy of patches as in classical hierarchical radiosity but with another refinement criterion T_P that uses a more precise form factor computation method. We use two separate thresholds since the refinement criterion used for facets and patches is not the same as the one used to link macrofacets.

In order to be more vigilant regarding the choice of the LOD approximation, we choose a small value of the threshold T_{LOD}. On the other hand, T_P may be chosen larger or equal to T_{LOD}.

In the same way, if the traversal of E's LOD hierarchy leads to a leaf P while F_{R-E} still remains higher than T_{LOD}, the same process is applied to P.
The algorithms describing our HR algorithm are given by figure 4. The following explains the role of functions invoked in these figures which are slightly different from the classical ones used in HR.

The function *Refine_And_Link()* recursively subdivides *either* the emitting macrofacet, facet or patch *or* the receiving macrofacet, facet or patch. (Object oriented languages, such as C++ and Java, enable us to handle easily these different data structures - this feature is called *polymorphism*). It establishes links when one of the refinement criterion is met. This criterion compares the computed form factor with the thresholds T_P or T_{LOD}.

Oracle(N_E,N_R) selects the node (N_E if $F_{R-E} > F_{E-R}$ else N_R) to be subdivided. This node could be a macrofacet, a facet or a patch.

Link() links together two nodes which could be macrofacets, facets or patches.

Shoot_One_Link() is a function which shoots the energy of the emitter through the links determined by the function *Refine_And_Link()*.

Shoot_Links(E) traverses, from the root (root macrofacet) to the leaves, both the LOD hierarchy and the hierarchy of patches associated with

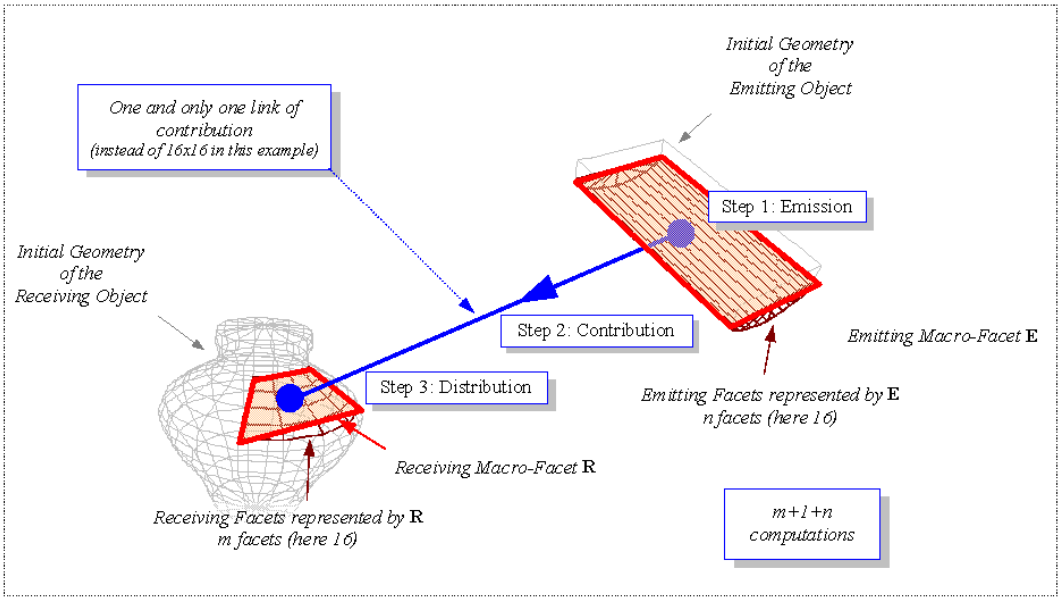

Figure 3: Energy exchange between two macrofacets

the emitting root macrofacet E. At each node N of these hierarchies, this function invokes the function Shoot_One_Link() to compute the contribution of each link between N and nodes belonging to other hierarchies.

Shoot_Energy(E) computes the radiosity B_{Gather} gathered by a receiving macrofacet or a facet or a patch when E emits its energy.

3 Energy Transfers between two HLODs

Let E and R be the emitting and receiving macrofacets respectively. Let us call e_k and r_k the children nodes of E and R in their respective LOD hierarchies, e_k and r_k may be facets or macrofacets.

3.1 Internal Shooting (*Step 0*)

Consider figure 5. Suppose a root macrofacet E has been chosen for shooting its energy. Before invoking the functions *RefineAndLink()* and *Shoot_Links()*, all the E's descendants must shoot their energy one to another according to one iteration of Jacobi type. To this end, the LOD hierarchy associated with E is traversed from the leaves to the root. At each depth level of this hierarchy:

- The nodes having the same parent shoot their energy one to another;
- Move to the upper depth level and repeat.

This energy transfer between the children nodes of the same parent will be called *Internal Shooting*. Its role is to avoid sub-estimating the flux emitted by a macrofacet and thus it guarantees the energy conservation.

Let us explain the internal shooting operation with the help of figure 5.

First, the facets e_j and e_k shoot their energy one to another and so do e_l and e_m, using classical HR. Then we compute the contextual radiosity of E_7 (see section 3.2) and shoot its energy to E_8 using the updated radiosities of e_j and e_k (see section 3.3). Then E_8 distributes its gathered energy to its descendants (say e_l and e_m) as seen in section 3.4. We proceed in the same way for shooting the energy of E_8 toward E_7. This operation is repeated for the other upper depth levels (i.e. coarser approximations) in the LOD hierarchy as explained by the algorithm of figure 4 and figure 5. To make the *Internal Shooting* faster, two nodes of different parents (such as e_j and e_l in figure 5) do not exchange energy directly but through their respective parents (E_7 and E_8). Note that, at the first call, the argument of *Internal_Shooting*() is a root macrofacet, say E in figure 5.

3.2 Emission (*Step 1, see figure 3*)

Using, as is, the unshot flux of the emitting root macrofacet E is imprecise. Instead, we propose to compute the flux of E that has to be actually considered for the energy transfer $E \rightarrow R$. This flux divided by the surface area of E will be called *contextual radiosity* (denoted B_c) from now on. Its particularity is that it accounts for the orientations of the nodes e_k, which makes more accurate the energy transfer between two nodes of LOD hierarchies.

The total flux of E is :

$$\Phi_E = A_E \cdot B_E, \tag{1}$$

where A_E and B_E are the surface area and the radiosity of E respectively.

```
HR_With_LOD(LOD_Criterion T_LOD, HR_Criterion T_P)
    {
    LOD_HR_Node E;
    While not convergence()
        {
        E = Root_Macrofacet_Of_Maximum_Flux() ;
        Internal_Shooting(E);
        for each root macrofacet R // An input polygon without LODs is also a root macrofacet
            {
            RefineAndLink(E, R, T_LOD, T_P) ; // T_LOD and T_P are form factor thresholds
            Shoot_Links(E) ;
            Remove_Links(E);
            for each leaf P of R's LOD hierarchy
                Push_Pull(P) ;
            }
        ΔΦ_E = 0 ; // Reset of the unshot flux of the emitting macrofacet
        }
    }

Internal_Shooting(LOD_HR_Node E, HR_Criterion T_P)
    {
    // Traverse the LOD Hierarchy of the the emitting macrofacet E
    for each child C_E of E
        if C_E is not a leaf of LOD hierarchy
            Internal_Shooting(C_E,T_P);

    // Energy exchange between nodes
    for each child C_E of E
        for each child C'_E of E (C_E ≠ C'_E)
            if C_E and C'_E are leaves of LOD hierarchy
                {
                // Classical HR
                Refine_And_Link(C_E, C'_E, 0, T_P) ;
                Shoot_Links(C_E) ;
                Remove_Links(C_E);
                Push_Pull(C_E);
                }
            else
                {
                // Energy exchange with LODs, only one link is created
                Link(C_E, C'_E);
                Shoot_Links(C_E);
                Remove_Links(C_E);
                }
    }

Refine_And_Link(LOD_HR_Node N_E, LOD_HR_Node N_R, LOD_Criterion T_LOD, HR_Criterion T_P)
    {
    OracleResult SubdivisionChoice=Oracle(N_E, N_R, T_LOD, T_P) ;
    // Uses T_LOD or T_P depending on the type of N_E and N_R (macrofacet or facet/patch)
    // In the first case, geometrical form-factor w/o visibility is used
    // In the second case, classical HR BF refinement is used

    switch(SubdivisionChoice)
        {
        Case None :
            Link(N_E, N_R) ; // Links N_E to N_R ; Both N_E to N_R could be macrofacets, facets or patches
        Case Emitter :
            Subdivide(N_E,ChildNodes) ; // Gives macrofacets, facets or patches children of NE
            For each node C_N in ChildNodes
                Refine_And_Link(C_N, N_R, T_LOD, T_P) ;
        Case Receiver :
            Subdivide(N_R,ChildNodes) ; // Gives macrofacets, facets or patches children of N_R
            For each node C_N in ChildNodes
                Refine_And_Link(N_E, C_N, T_LOD, T_P) ;
        }
    }

Shoot_One_Link(LOD_HR_Link L)
    {
    LOD_HR_Node N_E=Link.Emitter ;
    LOD_HR_Node N_R=Link.REceiver ;

    if(N_E is a MacroFacet and N_E is not a leaf of a LOD Hierarchy)
        B=Compute_Contextual_Radiosity(N_E, N_R) ; // Equation 6
    else
        B=Unshot_Radiosity(N_E) ;
    if(N R is a MacroFacet and N_R is not a leaf of a LOD Hierarchy)
        B_Gather= ShootEnergy(N_E) ; // Equation 7
        B=Distribute_Energy_To_Descendants(N_E, N_R, B_Gather) ; // Equation 11
    else
        B_Gather= Shoot_Energy(N_E) ; // Equation 7
    }
```

Figure 4: Hierarchical Radiosity with LODs

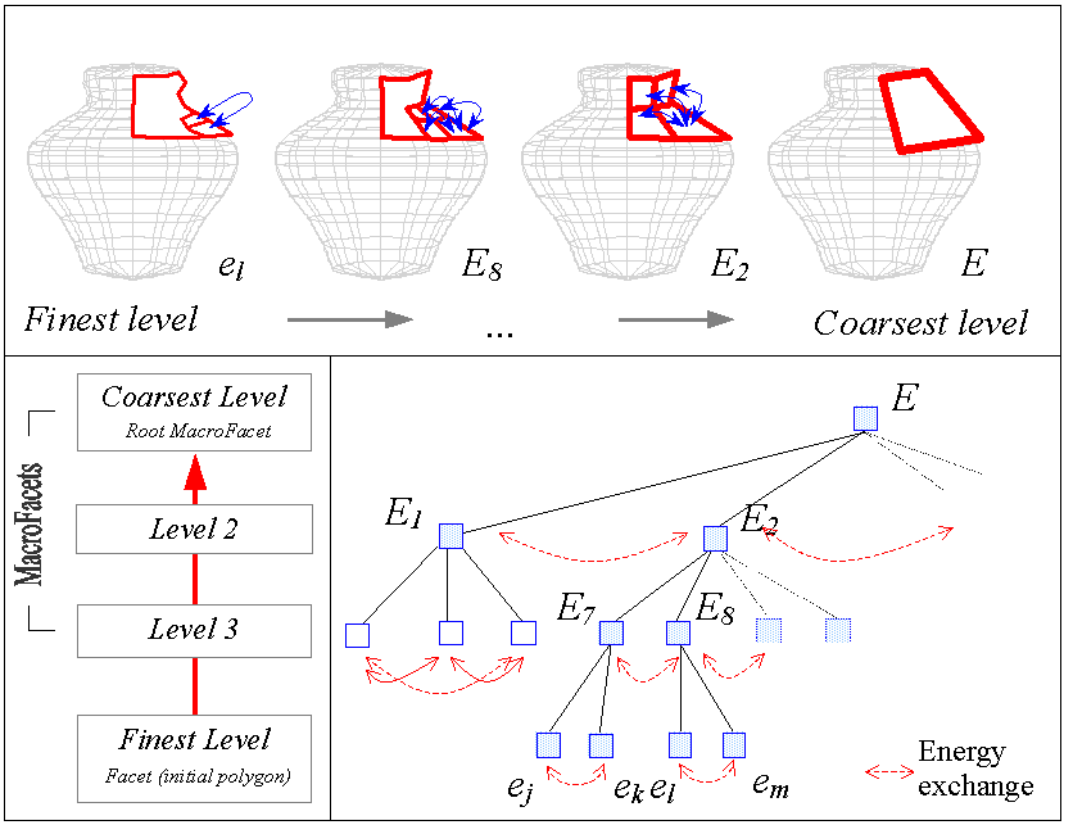

Figure 5: Internal Shooting

The flux $\Phi_{E\to R}$ emitted by E toward R is:

$$\Phi_{E\to R} = A_E \cdot B_c \cdot F_{E-R}, \tag{2}$$

F_{E-R} being the form factor between E and R.

Taking into account the orientations of the nodes e_k (facets or macrofacets), yields:

$$\Phi_{E\to R} = \sum_k \Phi_{e_k\to R} = \sum_k A_{e_k} \cdot B_{e_k} \cdot F_{e_k-R}, \tag{3}$$

As mentioned before, two nodes E and R of LOD hierarchies are linked only if the form factors (not including the visibility factor) F_{E-R} and F_{R-E} are below a threshold T_{LOD}. If T_{LOD} is small enough the F_{e_k-R} may be approximated as differential form factors. Consequently:

$$F_{e_k-R} = \frac{\cos\Theta_{e_kR} \cdot \cos\Theta_{Re_k} \cdot A_R}{\pi \cdot d^2_{e_k-R}}, \tag{4}$$

d_{e_k-R} being the distance separating the centers of e_k and R.

Similarly we can write:

$$F_{E-R} = \frac{\cos\Theta_{ER} \cdot \cos\Theta_{RE} \cdot A_R}{\pi \cdot d^2_{E-R}}, \tag{5}$$

From equations 2, 3, 4 and 5 we get:

$$B_c = \sum_k \frac{A_{e_k} \cdot B_{e_k} \cdot F_{e_k-R}}{A_E \cdot F_{E-R}} \approx \sum_k \frac{A_{e_k}}{A_E} \cdot \frac{\cos\Theta_{e_kR}}{\cos\Theta_{ER}} \cdot B_{e_k}, \tag{6}$$

assuming $d_{e_k-R} \approx d_{E-R}$ and $\Theta_{Re_k} \approx \Theta_{RE}$ $\forall k$ (see figure 6). B_c has to be multiplied by a visibility factor V_{e_kR} between e_k and R to take into account self-shadowing. We can observe that equation 6 accounts for the orientations of each node e_k relative to the location of the emitting macrofacet R, which makes more precise the radiosity emission.

3.3 Contribution (*Step 2, see figure 3*)

The radiosity due to E of the receiver node R (of a LOD hierarchy) is then:

$$B_R = \rho_R \cdot F_{R-E} \cdot B_c, \tag{7}$$

where ρ_R, the average reflectivity of R, is given by : $\rho_R = \sum_k \frac{A_{r_k}}{A_R} \cdot \rho_{r_k}$

Here F_{R-E} includes the visibility factor evaluated by tracing visibility rays between sample points on E and R.

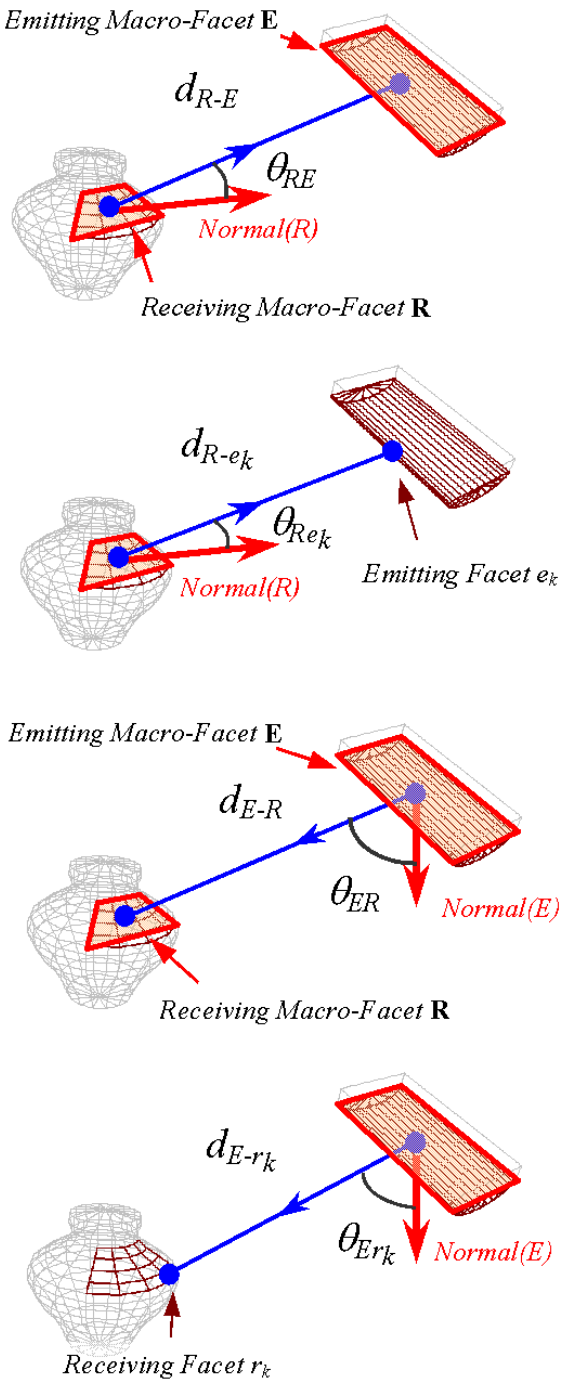

Figure 6: Angles

3.4 Distribution (*Step 3, see figure 3*)

In HR, the radiosity gathered by a facet is pushed/pulled through the associated hierarchy of patches (HP). As for a macrofacet R, whenever it gathers energy from E it distributes it to its descendants (macrofacets and facets) as explained hereafter. This energy distribution must account for the location of E and the orientations of each child node r_k of R.

Recall that B_R is the radiosity gathered by R and due to the emitter E. Let us denote B_{r_k} the radiosity that has to be distributed to each r_k by its parent macrofacet R. We can write:

$$A_R \cdot B_R = \rho_R \cdot F_{R-E} \cdot B_c \cdot A_R = \rho_R \cdot F_{E-R} \cdot B_c \cdot A_E \quad (8)$$

$$A_{r_k} \cdot B_{r_k} = \rho_{r_k} \cdot F_{r_k-E} \cdot B_c \cdot A_{r_k} = \rho_{r_k} \cdot F_{E-r_k} \cdot B_c \cdot A_E \quad (9)$$

As before we suppose T_{LOD} small enough for using the differential form of F_{E-r_k}. We get then:

$$F_{E-r_k} = \frac{\cos\Theta_{Er_k} \cdot \cos\Theta_{r_kE} \cdot A_{r_k}}{\pi \cdot d^2_{E-r_k}}, \quad (10)$$

By dividing equations 8 by 9, substituting equation 10, assuming $\Theta_{Er_k} \approx \Theta_{ER}, d_{E-r_k} \approx d_{E-R}, \forall k$ (see figure 6) and adding the visibility factor V_{Er_k}, we obtain:

$$B_{r_k} \approx B_R \cdot \frac{\cos\Theta_{r_kE}}{\cos\Theta_{RE}} \cdot \frac{\rho_{r_k}}{\rho_R} \cdot V_{Er_k} \quad (11)$$

We see that equation 10 accounts for the orientations of the nodes r_k and the spatial location of the emitter E, which makes more precise the radiosity distribution. Note that the equations 6, 7 and 11, provide us with simple and fast expressions giving the contextual and distributed radiosities. Indeed, suppose that E and R have respectively m and n children. Only $(m+1+n)$ form factors F_{e_k-R} and F_{E-r_k} are computed (including visibility), avoiding then the evaluation of $m*n$ form factors $F_{e_k-r_k}$.

4 Results and discussion

As we will demonstrate in this section, the main advantages of our method consists in well handling complex light sources as well as complex objects.

We have tested our algorithm with three different scenes under different conditions (by varying the number of light sources, the number of objects, their LODs approximation, etc), producing a total of 14 distinct test scenes. All these scenes are made up of different kinds of objects approximated with LODs. Scene 1, called '*sofas*', is a classical test scene representing a room furnished with several sofas and

Scene	'Sofas'	'Sofas'	'Sofas'	'Objects'	'Objects'	'Objects'
Configuration	3 sofas	4 sofas	4 sofas and more	1 object (*vase*)	3 objects	5 objects
Number of input polygons	493	1011	1451	4246	12438	20630
Number of light sources	1	1	1	4	4	4
Number of emitting polygons	32	32	32	128	128	128
Full solution after convergence	11.03 s	11.21 s	11.46 s	7.86 s	39 s	77 s

Table 1: Some computing times for scenes 1 and 2

Scene	'Objects'	'Objects'	'Objects'	'Objects'	'Objects'	'Objects'
Configuration	1 vase 1 light	1 vase 2 lights	1 vase 4 lights	coarse HLOD	medium HLOD	fine HLOD
Number of input polygons	4135	4172	4246	295	1063	4135
Number of root macrofacets (Vase)	16	16	16	16	16	16
Number of lights sources	1	2	4	1	1	1
Number of emitting polygons	32	64	128	32	32	32
Full solution after convergence	4.17 s	6.24 s	7.86 s	24 s	31 s	58 s

Table 2: Two case studies

Scene	'Virtual University'	'Virtual University'	'Virtual University'
Configuration	10K	20K	40K
Number of input polygons	10600	21200	42400
Number of lights sources	23	46	92
Number of emitting polygons	736	1472	2944
Direct lighting	1 m 16 s	4 m 20 s	16 m 19 s
80 percent convergence	2 m 33 s	8 m 43 s	32 m 51 s
Full solution after convergence	6 m 28 s	21 m 46 s	1 h 23 m 28 s

Table 3: Progressive results on 'virtual university'

Scene	'Sofas'	'Objects'	'Virtual University'
Configuration	3 sofas	1 vase & 1 light	10K
Number of light sources	1	1	23
Number of emitting polygons	16	16	736
HR + Clustering, after the 1^{st} Iteration	1 m 36 s	51 s	34 m 43 s
HR + Clustering , after the 2^{nd} Iteration	4 m 3 s	1 m 44 s	2 h 38 m 24 s
HR with LODs after convergence	11.03 s	7.86 s	6 m 28 s

Table 4: Computing times for three different scenes with a clustering radiosity algorithm

lit with only one light source. Scene 2, called '*objects*', contains more complex curved objects (as seen in figure 9). These objects have been generated with *NURBS* surfaces tesselated into polygons. And finally, Scene 3, called '*virtual university*', is an architectural scene modeled with rooms and hallways (see figures 10, 11 and 12).

The number of input polygons, N_{IP}, is equal to the number of polygonal surfaces at the finest level of representation. The number of root macro-facets N_{RM} (the polygonal surfaces at the highest level of representation) could be far below N_{IP}.

We define σ as the ratio N_{IP}/N_{RM}. The higher the value of σ, the lower is the number of root macrofacets with respect to the number of input polygons. A polygon belonging to a light source is counted as an *emitting polygon*.

Table 1 shows some run-times for scenes 'sofas' and 'objects'. Table 2 presents results for two case studies we have made with the second scene ('objects'). The first study (the three left most columns) shows the advantage of our method for scenes where the geometry of a light source is complex (the light source here is represented with thirty two input polygons). As more and more light sources are added, the computing times increase linearly (the ratio N_{IP}/computing-time is almost constant). As for the second study, we tried different values for σ (which means that the vase is represented with a finer and finer tesselation but N_{RM} stays constant). Here again, the ratio N_{IP}/computing-time is almost constant. This shows that the effectiveness of the algorithm is not only related to the depth of each LOD hierarchy but also to its breadth that varies as σ.

The data in tables 1 and 2 confirm that the average complexity of the algorithm is O(m+1+n) for energy

transfers involving HLODs (cf section 3).
Table 3 gives the computing times for the architectural scene (scene 3, the 'virtual university') with an increasing number of rooms. This table presents some times obtained at different stages (direct lighting only, 80 % of convergence, full convergence). Under these conditions, the direct lighting phase takes one fifth of the full solution time and the 80 % convergence stage is reached after less than half of the full solution time ($\approx$ 40 %). This behavior shows that the algorithm should be quite advantageous in modeling applications where users need fast scene previews. The results of this test are also shown in the figure 7 (plot 1). The second plot presented in this figure shows a comparison between our algorithm and a classical HR. The speed-ups obtained are significant and encouraging (roughly ranging from 10 to 110). The speed-ups are greatest when scenes have large numbers of emitting polygons and many complex objects, a situation that is difficult for classical HR. The smallest speed-ups are obtained when large surfaces such as walls, floors or ceilings have to reflect a large amount of energy. Exchanges between such surfaces do not involve any LOD representation and increase the computing time and memory used.

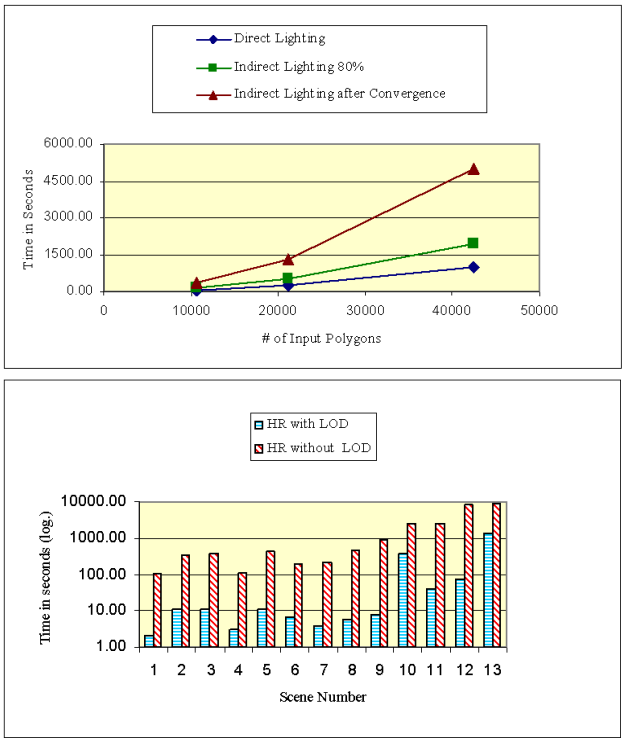

Figure 7: plot 1 (left): computing times for 'Virtual University', plot 2 (right): comparison with classical HR

Finally, table 4 shows some results for comparing our method with a HR algorithm using clustering. Here, the comparison can only be qualitative since it is not obvious to compare these two different approaches with two different implementations (e.g spectral versus rgb color representation, shooting method versus gathering method, different refinement criteria, different visibility computation methods, etc). Note that, after two iterations only, the HR algorithm with clustering takes more time compared to our HR with LODs. This explains the results given in table 4.

Our method is not only a good alternative to clustering techniques but is also a perfect complement to it. We did not have enough time to implement a version using both approaches but our speculation is that an ideal hierarchical solution might be: Partitioning + clustering + LODs (Macrofacets) + HR with facets and patches.

In terms of memory usage the LODs do not give any significant improvement since exchanges between walls, floors and ceilings involve the highest number of links created (with no use of any HLOD). Therefore, our algorithm requires the same amount of memory as any classical HR. Clustering techniques also do not handle this case properly.

5 Images

Figure 8 presents a version of the scene 'objects' with one light source and one object (a vase). This object is lit from above by a light source also represented with a LOD hierarchy. In this scene, the σ value for the vase is important and equal to 256.

This image shows that our algorithm is able to render complex geometries that are not well handled by classical HR algorithms. There is no noticeable difference between the results provided by the classical HR and our method (which in this case, required 30 times less computational time). Note that shading and self-shadowing are well represented in these images.

Figure 9 presents a view and close-up for a different configuration of this scene (in this case, it contains 5 objects and 4 colored light sources, all represented with LODs). The accuracy of the method is illustrated by shadows on the floor which respect the curvature of these objects.

In Figure 10, the scene 'virtual university' is shown. The left most image has been generated by our algorithm while the right most one has been obtained with a clustering technique (this configuration (10K) contains ten thousand input polygons and 736 emitting polygons). Few differences are visible and most of them can be attributed to the different color representations used by the two algorithms (spectral in our implementation versus RGB).

Finally, figures 11 and 12 shows rendering of the 'virtual university' at its highest level of complexity (forty thousand input polygons including 2944 emitting ones).

6 Conclusion and future work

We have proposed a global illumination method combining hierarchical radiosity and LOD approximations of object surfaces. Our approach accounts more precisely for the orientations of the children nodes for both the emitting and receiving macrofacets. When selecting a macrofacet for shooting, internal shooting followed by the computation of the associated contextual radiosity make the energy emission more precise. A macrofacet may be linked to another macrofacet, a facet or patch. The shooting process used depends on the linked primitives.
The different cases studied during this research tend to prove that LOD representations are well suited for complex geometry and complex light sources. As mentionned before, a global framework including partitioning techniques, clustering methods, LODs representation and classical radiosity is probably an excellent architecture for future implementation. It should be well-suited for any kind of geometry (e.g. [4] presents a framework, developped at IRISA, where clusters are used to speed up the visibility computation involved by a partitioning technique).
One way to extend this work is to generate each HLOD on the fly. However, as we use a bottom-up approach (the input polygons represent the actual geometry) it is not easy to maintain the hierarchy during the computation process. Another way to extend this work is to handle non-diffuse surfaces with arbitrary reflectances. This could be done by using extended three-point transport as presented in [23].

7 Acknowledgments

The authors would like to thank iMagis who left at our disposal one version of *GIS*, a clustering algorithm they developed. Many thanks to Xavier Granier and Georges Drettakis for their help and their useful discussions.
The authors also would like to thank Carine Clavaud, for her help to this project and Jim Ferwerda for his proof-reading.

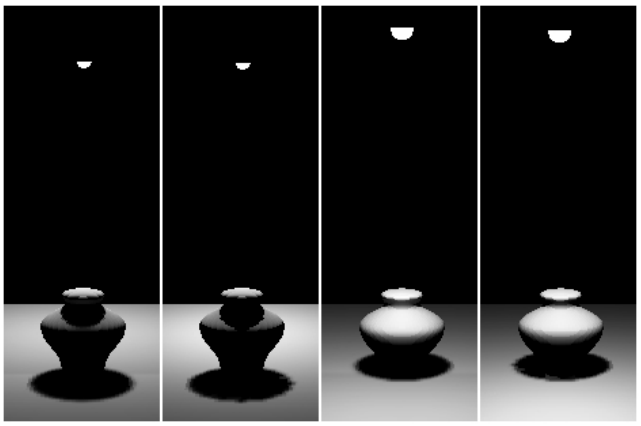

Figure 8: Scene 2 - 'Objects' - Configuration 1 object (*the vase*) & 1 light source - Comparison between radiosity with LODs and without LOD, 1: w/o LOD rear view, 2: w/ LOD rear view, 3: w/o LOD front view, 4: w/ LOD front view

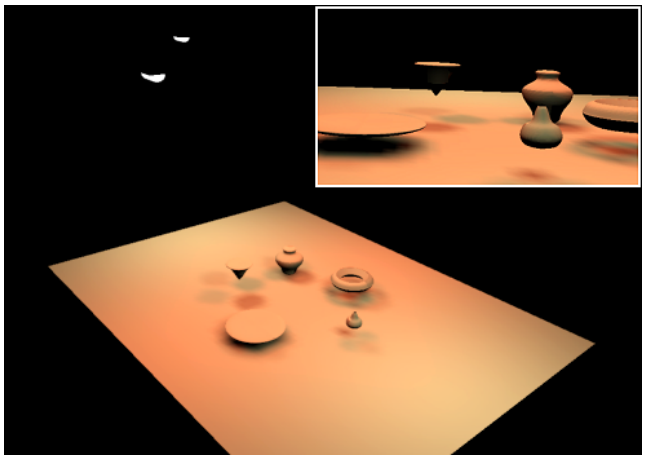

Figure 9: Scene 2 - 'Objects' - Configuration 5 objects & 4 colored light sources - Full view and close up

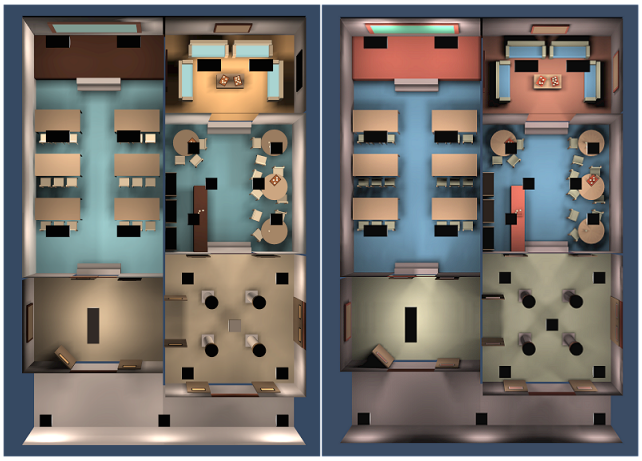

Figure 10: Scene 3 - Configuration 10K - Comparison between radiosity with LODs (left) and with clusters (right)

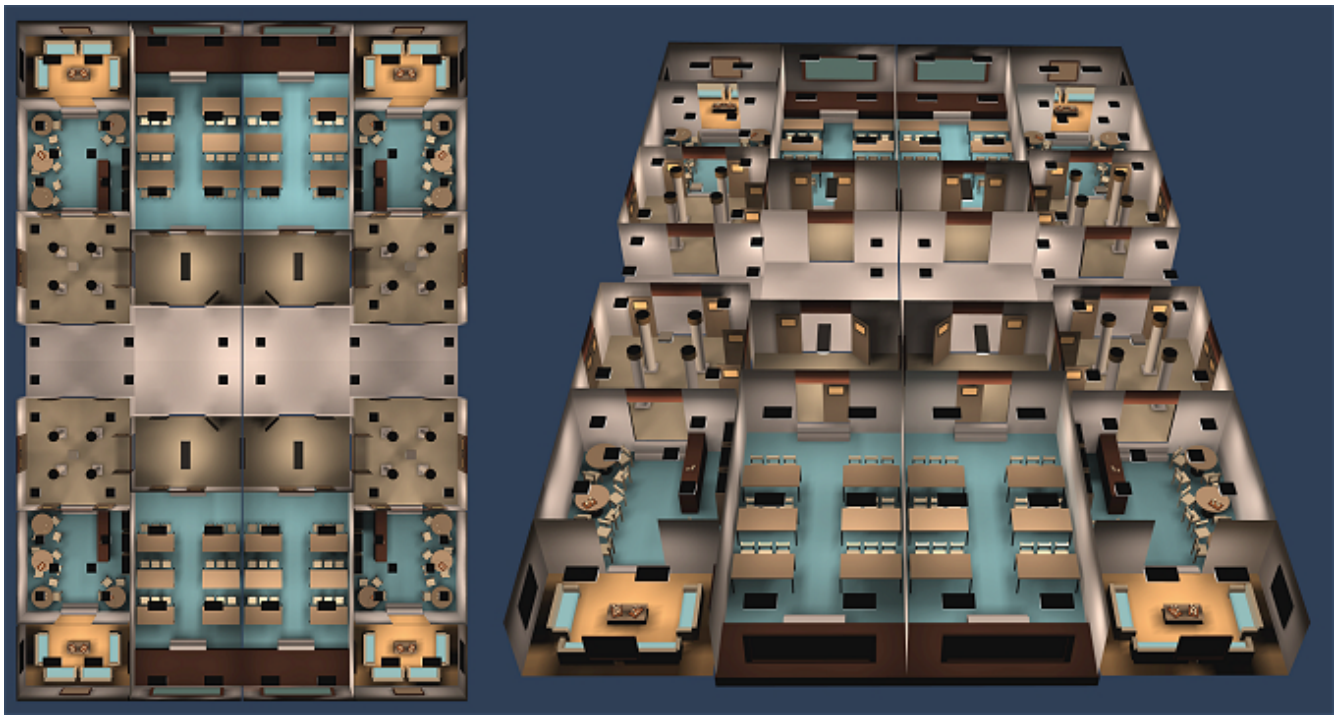

Figure 11: Scene 3 - 'Virtual University' - Configuration 40K - Panoramas

Figure 12: Scene 3 - 'Virtual University' - Configuration 40K - Close-ups

References

[1] Hanrahan P. Salzman D. Aupperle A. A rapid hierarchical radiosity algorithm for unoccluded environments. *Computer Graphics*, 25(4):197–205, July 1991.

[2] Airey J.M. Increasing update rates in the building walkthrough system with automatic model-space subdivision and potentially visible set calculation, phd thesis. Technical Report TR 90-027, University of north Carolina at Chapel hill, 1990.

[3] Funkhouser T. Teller S. Sequin C. Khorramabadi D. The uc berkeley system for interactive visulalization of large architectural models. *Presence*, 5(1):13–44, 1996.

[4] Meneveaux D. Bouatouch K. Maisel E. Delmont R. A new partitioning method for architectual environments. *Journal of Visualization and Computer Animation*, 9:195–213, 1998.

[5] Teller S. Fowler C. Funkhouser T. Hanrahan P. Partitioning and ordering large radiosity computations. In *Computer Graphics Proceedings, Annual Conference Series*, pages 443–450, 1994.

[6] Smits B. Arvo J. Greenberg D. A clustering algorithm for radiosity in complex environments. In *Computer Graphics Proceedings, Annual Conference Series*, pages 435–442, 1994.

[7] Sillion F. Clustering and volume scattering for hierarchical radiosity calculations. *5th Eurographics Workshop on Rendering*, 1994.

[8] Willmott A. Heckbert P. Garland M. Face cluster radiosity. In *10th Eurographics Workshop on Rendering*, pages 293–304, Granada, Spain., 1999.

[9] Rushmeier H. Patterson C. Veerasamy A. Geometric simplification for indirect illumination calculations. In *Graphics Interface*, Canada, 1993.

[10] Kok A. J. F. *Ray Tracing and Radiosity Algorithms for Photorealistic Image Synthesis.* Delft University Press, 1994.

[11] Gibson S. Hubbold R.J. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15:297–310, 1996.

[12] Sillion F. Drettakis G. Feature-based control of visibility: A multiresolution clustering algorithm for global illumination. In *Siggraph'95.* ACM, 1995.

[13] Sillion F. Drettakis G. Soler C. A clustering algorithm for radiance calculation in general environments. In *6th Eurographics Workshop on Rendering*, June 1995.

[14] Christensen P.H. Lischinski D. Stollnitz E. Salesin D. Clustering for glossy global illumination. *ACM Transaction On Graphics*, 16(1):3–33, January 1997.

[15] Stamminger M. Slusallek P. Siedel H.P. Three point clustering for radiance computations. Springer Verlag, Rendering Techniques (Proceedings of the 9th Eurographics Workshop on Rendering), 1998.

[16] Hasenfratz J-M. Damez C. Sillion F. Drettakis G. A practical analysis of clustering strategies for hierarchical radiosity. In *Eurographics'99 Main Conference, Vol 18, Num 3*, 1999.

[17] Varshney A. Hierarchical geometric approximations. Technical Report 27599-3175, Department of Computer Science, University of North Carolina at Chapel Hill, 1994.

[18] Heckbert P. Garland M. Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon University Technical Report, 1997.

[19] Garland M. Heckbert P. Surface simplification using quadric error metrics. In *Siggraph*, 1997.

[20] Hoppe H. DeRose T. Duchamp T. Mc-Donald J. Stuetzle W. Mesh optimization. In *Computer Graphics Proceedings, Annual Conference Series*, pages 19–26, August 1993.

[21] Hoppe H. Progressive meshes. In *Siggraph*, pages 99–108, 1996.

[22] Rossignac J. Borrel P. Multi-resolution 3d approximation for rendering complex scenes. In *Second Conference on Geometric Modelling*, pages 453–465, 1993.

[23] Dumont R. Bouatouch K. Gosselin P. A faster progressive algorithm for glossy global illumination. *Computer Graphics Forum*, 1, March 1999.

[24] Stamminger M. Schirmacher H. Slusallek P. Seidel H.P. Getting rid of links in hierarchical radiosity. In *Proceedings of Eurographics Annual conferences*, 1998.