A Clustering Algorithm for Radiance Calculation In General Environments

François Sillion, George Drettakis* , Cyril Soler

iMAGIS **

Abstract: This paper introduces an efficient hierarchical algorithm capable of simulating light transfer for complex scenes containing non-diffuse surfaces. The algorithm stems from a new formulation of hierarchical energy exchanges between object clusters, based on the explicit representation of directional radiometric distributions. This approach permits the simplified evaluation of energy transfers and error bounds between clusters. Representation and storage issues are central to this type of algorithm: we discuss the different choices for representing directional distributions, and the choice between explicit storage or immediate propagation of directional information in the hierarchy. The framework presented is well suited to a multi-resolution representation, which may in turn significantly alleviate the storage problems. Results from an implementation are presented, indicating the feasibility of the approach and its capacity to treat complex scenes.

1 Introduction

The hierarchical radiosity algorithm permits the efficient computation of radiosity solution within well-understood error-bounds. Its main limitation is the "initial-linking" step, which for scenes of diffuse polygons adds a quadratic computational cost. As a consequence the algorithm is unusable for large environments. Recently presented clustering algorithms for hierarchical solutions [10, 6], avoid the quadratic cost by first clustering the environment and then refining the clusters.

Nonetheless, little work has been performed for non-diffuse environments. Two-pass algorithms [9, 11] and a general solution using directional representations [7] have treated more general environments in the context of progressive refinement radiosity. A hierarchical solution to general environments has also been proposed [1], but in the case of that algorithm the initial linking cost becomes $\mathcal{O}(n^3)$ in the number of initial polygons, making it unusable even for moderately complex scenes.

The processing of complex environments with general reflectors is a necessity, since almost all interesting scenes contain at least some percentage of non-diffuse materials. In this paper we present a framework which provides the necessary machinery for the treatment of non-diffuse environments in the context of a hierarchical clustering algorithm. This framework is a natural extension of previous clustering methods since, as noted before [6], clusters do not behave as isotropic scatterers, even if composed solely of diffuse surfaces. It is based on the representation of radiant intensity by directional distribution functions, and extends the spirit presented in [7] to hierarchical clustering. The result is the first efficient hierarchical algorithm permitting the efficient of complex, non-diffuse environments. In addition, this representation affords a smooth transition between the representation at the level of (non-diffuse and diffuse) surfaces to the

---

* The second author performed this research with an ERCIM fellowship (funded by the EU Commission), partially at UPC, Barcelona, Spain and GMD, St. Augustin, Germany.

** iMAGIS is a joint research project of CNRS/INRIA/INPG/UJF. Postal address: B.P. 53, F-38041 Grenoble Cedex 9, France. Contact E-mail: `Francois.Sillion@imag.fr`.

level of clusters. Finally, the framework opens the way to an efficient multi-resolution representation of light properties for clusters.

In contrast with previous clustering approaches our new method is based on the storage of directional properties with the clusters. This approach requires the reconsideration of some of the quantities previously used since we are now dealing with directional energy exchanges between clusters. In Section 2 we characterise the directional properties of clusters which are used in our solution. In Section 3 we introduce the new algorithm which is based on the directional representation, in Section 4 we discuss the issues pertaining to possible approaches to storing directional distributions and in Section 5 we present some implementation issues and some first results. We conclude in Section 6 with a discussion of limitations and the directions for future research.

2 Characterization of directional energy transfer

As outlined above, we will be treating the light leaving and impinging on clusters as a function of direction. In particular we want to be able to store and manipulate directional functions to characterize the radiant behaviour of a cluster. In this section we discuss the physical quantities used, their representation and their relation to traditional radiosity variables.

For the most general discussion of directional light transfer, we consider light leaving the cluster, light impinging on the cluster, and light passing through the cluster. We also introduce a particular directional function useful for the expression of energy exchanges with distributions. In the remainder of this paper we will denote a direction in space by a unit vector, with the convention that $\vec{u}$ represents an outgoing direction and $\vec{v}$ an incident direction (See Fig. 1).

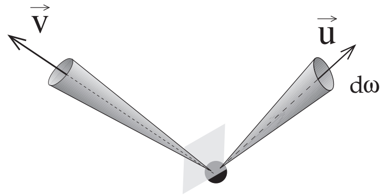

Fig. 1. Notations used for directional functions.

2.1 Outgoing Light

For the description of light leaving the cluster, we use *radiant intensity*, I, representing power per unit solid angle. At a point x on a surface, radiant intensity is related to radiance by the following formula:

$$dI(x, \vec{u}) = L(x, \vec{u})\, dA\, (\vec{u} \cdot \vec{n})\,, \tag{1}$$

where $\vec{n}$ is the surface normal and dA is the differential surface area around point x. In the case of a diffuse surface with radiosity B, radiant intensity is thus given by

$$dI(x, \vec{u}) = \frac{B}{\pi}\, dA\, (\vec{u} \cdot \vec{n})\,.$$

2.2 Incoming Light

For light arriving on a cluster, we use the standard (incoming) *radiance* quantity, defined as the amount of power received per unit area perpendicular to the direction of incidence and per unit solid angle.

With this definition, if the distribution of incident radiance at point x is $E(x, \vec{v})$, the incoming flux density per unit solid angle on a surface placed at x with normal direction $\vec{n}$ is

$$E_s(x, \vec{v}) = E(x, \vec{v})\ (\vec{v} \cdot \vec{n}) \tag{2}$$

2.3 The Tangent-sphere function

In Equations 1 and 2 above, the scalar products must be understood as being zero if the surface is not facing the right direction. For notational convenience we represent this extended scalar product as a function of $\vec{u}$. Let us define the *tangent-sphere* function $T_{\vec{n}}(\vec{u})$ for a direction $\vec{n}$ by

$$T_{\vec{n}}(\vec{u}) = \begin{cases} \vec{u} \cdot \vec{n} & \text{if } \vec{u} \cdot \vec{n} \geq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

As shown in Fig. 2 the surface given in spherical coordinates by $r = T_{\vec{n}}(\vec{u})$ has the shape of a sphere tangent to the plane orthogonal to $\vec{n}$.

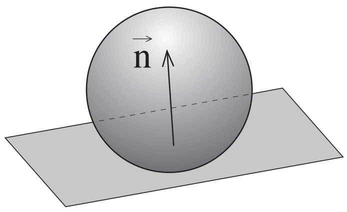

Fig. 2. Tangent-Sphere function.

Using this function, Equations 1 and 2 can be rewritten as

$$I(x, \vec{u}) = L(x, \vec{u})\ dA\ T_{\vec{n}}(\vec{u}) \tag{4}$$

and

$$E_s(x, \vec{v}) = E(x, \vec{v})\ T_{\vec{n}}(\vec{v}) \tag{5}$$

2.4 Extinction properties

The transmission properties of object clusters can be discussed using a fruitful analogy with semi-transparent volumes with optical extinction properties. Previous work along this line has proposed to compute equivalent isotropic extinction coefficients for object clusters based on the total area they contain [6] ($\kappa = A/4V$, where A is the total surface area of the objects in the cluster and V is its volume).

In the general approach presented here we lift the isotropic assumption and compute for each cluster a directional extinction coefficient, used to evaluate the attenuation of a light beam traversing the cluster in a given direction. The total projected area in a given direction can be precomputed and stored with each cluster. It is given by the following sum over the surfaces contained in the cluster:

$$\mathcal{A}(\vec{v}) = \sum_i A_i T_{\vec{n}_i}(\vec{v}) \tag{6}$$

A directional extinction coefficient is then obtained with the following formula:

$$\kappa(\vec{v}) = \frac{\mathcal{A}(\vec{v})}{V} \tag{7}$$

$\kappa(\vec{v})$ is used as in [6] to compute approximate transmission through a cluster, as it represents the rate of attenuation per unit length in the direction of interest. Note that the factor of 4 from the isotropic formula is no longer present, since it accounted for the averaging over all directions. Plate 1 (see Appendix) shows results obtained with directional extinction.

2.5 Light Scattering

For now we only consider the transformation of incoming light into outgoing light to take place at surfaces. We assume that a surface oriented in direction $\vec{n}$ is placed at the origin. The surface is small enough for all distributions to be safely assumed constant across its surface. One difficulty in expressing the general light scattering equation is that surface scattering is best described in a coordinate system that is local to the surface. Let us define a linear transformation $\boldsymbol{M}_{\vec{n}}$ such that $\vec{u}' = \boldsymbol{M}_{\vec{n}}\vec{u}$ is the unit vector representing the direction of $\vec{u}$ in a coordinate system attached to the surface. As shown in Fig. 3, both vectors are aligned, they simply have different coordinates because they are expressed in different frames of reference.

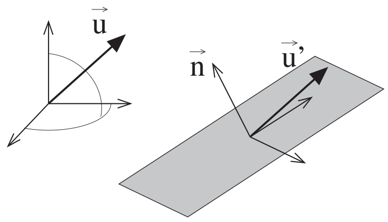

Fig. 3. Notations for the scattering equation.

Surface scattering In this paragraph we express all directions in the surface coordinate system. The radiance leaving the surface in a direction $\vec{u}'$ is given by

$$L(\vec{u}') = \int_{\vec{v}' \in \Omega_+} E_s(\vec{v}')\rho_{bd}(\vec{u}', \vec{v}')d\omega_{\vec{v}'} \tag{8}$$

where $E_s(\vec{v}')d\omega_{\vec{v}'}$ is the incident flux density on the surface from the differential solid angle $d\omega_{\vec{v}'}$ around direction $\vec{v}'$. Ω_+ is the upper hemisphere (above the surface).

Expressing radiant intensity from incident radiance We now wish to express the scattering equation using the directional quantities defined above, and in a general (world) coordinate system, not tied to any particular surface. This simply requires a number of coordinate transformations using $\boldsymbol{M}_{\vec{n}}$. Combining Equations 4 and 8, we can express the radiant intensity leaving a surface in direction $\vec{u}$ as

$$
\begin{aligned}
I(\vec{u}) &= A\, T_{\vec{n}}(\vec{u})\; L(\vec{u}) && (9)\\
&= A\, T_{\vec{n}}(\vec{u})\; L(\boldsymbol{M}_{\vec{n}}\vec{u}) && (10)\\
&= A\, T_{\vec{n}}(\vec{u}) \int_{\vec{v}'\in\Omega_+} E_s(\vec{v}')\rho_{bd}(\boldsymbol{M}_{\vec{n}}\vec{u},\vec{v}')d\omega_{\vec{v}'}\;. && (11)
\end{aligned}
$$

Using Equation 5 and changing the integration variable to be a unit vector in the hemisphere above the oriented surface, $\vec{v} = \boldsymbol{M}_{\vec{n}}^{-1}\vec{v}'$, we have

$$
I(\vec{u}) = A\, T_{\vec{n}}(\vec{u}) \int_{\vec{v}\in\boldsymbol{M}^{-1}\Omega_+} E(\vec{v})T_{\vec{n}}(\vec{v})\rho_{bd}(\boldsymbol{M}_{\vec{n}}\vec{u},\boldsymbol{M}_{\vec{n}}\vec{v})d\omega_{\vec{v}} \qquad (12)
$$

Ideal diffuse case For ideal diffuse surfaces, the BRDF is a constant, and Equation 12 reduces to

$$
I(\vec{u}) = A\, T_{\vec{n}}(\vec{u})\, \frac{\rho_d}{\pi} \int_{\vec{v}\in\boldsymbol{M}^{-1}\Omega_+} E(\vec{v})T_{\vec{n}}(\vec{v})d\omega_{\vec{v}} \qquad (13)
$$

The integral in Equation 13 represents the total incident flux density (irradiance) on the surface.

3 A Cluster-Based Illumination Algorithm for General Scenes

Existing radiosity clustering algorithms can be adapted to work with directional information, with little modification as described in this section. We assume here that the reader is familiar with hierarchical radiosity and clustering algorithms [3, 10, 6]. In these methods, a hierarchical subdivision structure of 3D space is used to collect surfaces into clusters. The main idea of the new general clustering algorithm is to associate to each cluster or surface a number of directional distributions representing its radiant properties. The scattering equation (12) must then be evaluated for each surface, using the appropriate incident radiance and radiant intensity distributions.

3.1 Form factor

Since we are using a radiant intensity distribution on the emitter, the estimation of energy transfer between a pair of objects is slightly different than with usual radiosity. Transfer estimates are needed in two stages of a hierarchical radiosity algorithm. First, a bound on the total energy transfer between two objects (or clusters) must be computed during the link refinement stage. Second, the actual energy transfer takes place in a *gathering* stage, where the incoming energy is computed across each link.

The notion of "form factor" used in our algorithm is redefined from purely algorithmic considerations: the form factor associated to each link is the scalar quantity by which the radiant intensity value of an emitter must be multiplied to obtain the incident irradiance (power per unit area perpendicular to the direction of propagation) on the receiver.

This quantity is simply derived from the expression of radiant intensity and irradiance, and is

$$F_{pq} = \int_p \int_q \frac{1}{r^2} dp\, dq \tag{14}$$

3.2 Link refinement

For the purpose of making a refinement decision, a hierarchical subdivision criterion must be defined. Our preliminary implementation uses an estimate of the energy transferred between two objects q and p (objects can be surfaces or clusters [6]). To obtain this estimate we select two sample points in p and q, yielding a direction $\vec{u}$. Multiplying $I_q(\vec{u})$ with the "form-factor" F_{pq} we obtain an incident irradiance contribution on p from direction $\vec{v} = -\vec{u}$, denoted by $\mathcal{E}_{pq}$. Note that, in a manner similar to Lischinski *at al*'s work [5], an actual bound on this transfer can be computed, provided we store not only the average radiant intensity but also the maximum radiant intensity for each object. To obtain an energy value from incident irradiance requires a multiplication by the total projected area of the cluster's contents in direction $\vec{v}$, $\mathcal{A}(\vec{v})$, introduced in Section 2.4. Our estimate of the energy contribution of the link between q and p is thus

$$P = \mathcal{A}_p(\vec{v})\mathcal{E}_{pq} \tag{15}$$
$$= \mathcal{A}_p(-\vec{u})I_q(\vec{u})F_{pq} \tag{16}$$

Note that the previous discussion ignores intra-cluster visibility issues. These are not treated in this paper, although recent work shows that it is possible to integrate their effect with reasonable cost [8]. It is interesting to note the benefit of storing the radiant intensity in the form of a directional distribution, since the transfer estimate does not require the interrogation of the cluster contents. This represents a potential gain over previous hierarchical clustering algorithms [6, 10].

3.3 Gather

Due to the change in quantities used to represent and store light, the traditional process of gathering across linked clusters or surfaces must be appropriately modified.

One of the most important choices to be made when representing directional properties in a hierarchy of clusters, is which properties to store explicitly at all levels in the hierarchy and which to store implicitly by pushing them down to the level at which additional storage cost is incurred. In particular the efficient treatment of incident energy contributions requires some attention. We consider here two alternatives, and discuss their relative merits.

Storing an incident radiance distribution The simplest directional clustering algorithm is probably one where incoming radiance is stored with each cluster, together with (outgoing) radiant intensity. The main advantage of this approach is that the amount of work performed for each link in the gathering phase is fixed, and does not depend on the clusters' complexity. This "constant-time" transfer computation, combined with the linear number of links with respect to the total number of surfaces [3, 10], results in a clustering algorithm with linear asymptotic complexity.

Unfortunately, storing incoming radiance is difficult and expensive. First,in the context of our framework we want to use a continuous, directional function representation. Incoming radiance is inherently discontinuous, as for instance the contribution of a given source is non-zero only for directions reaching the source. This difficulty can be eliminated by estimating a continuous approximation to each source's contribution to the incident radiance.

Consider again the transfer from q to p. Since our refinement criterion has established the link at this level, it is reasonable to assume that the transfer is well represented by a point-to-point calculation. An estimation of the error incurred by this assumption must evidently be undertaken in the future. The incident *irradiance* on p is obtained as explained in Section 3.2. This irradiance can be spread across the solid angle subtended by q, using a simple parametric filter in the shape of a peak. We are investigating the use of rotated $\cos^n(\theta)$ distributions as convolution filters. Clearly however this operation involves a significant additional computational cost.

In addition, explicit storage implies the need for an expensive convolution operation when pushing the incoming radiance down the hierarchy of clusters. At the transition from clusters to surfaces the conversion from incident radiance to radiant intensity must be performed, as shown in Equations 12 for the general case and 13 for the diffuse case. Again this implies significant additional computation.

Immediate propagation of incoming contributions An alternative to storage of incoming radiance is to explicitly push incoming light down the hierarchy at each gather operation. To perform this we no longer consider radiance, but *irradiance*, computed as in Section 3.2. This quantity, accompanied by the incoming direction $\vec{v}$ is pushed down the cluster hierarchy by simple addition. This irradiance is the term $E(\vec{v})d\omega_{\vec{v}}$ in Equation 12. At the surface level we need only evaluate Equation 12, replacing the integral by an "impulse" from direction $\vec{v}$, with the surface irradiance value $\mathcal{E}_{pq}T_{\vec{n}}(\vec{v})$. This surface irradiance is used to scale the surface's BRDF, which reduces to a constant for diffuse surfaces.

3.4 Push/Pull

In our implementation we have chosen the option of immediate pushing of incoming radiance as opposed to storing the quantity as a directional function. Thus the traditional Push-Pull operation only needs to perform the "Pull" portion, since the "Push" occurs at the gathering stage. Since radiant intensity is a power quantity, the radiant intensity of a cluster is obtained from that of its sub-clusters by simple summation. The result is a combined directional function representing the total radiant intensity of the cluster.

4 Representation of Directional Distributions

Several storage schemes have been investigated in the context of simulating non-diffuse radiant exchanges. A major difficulty in selecting a representation is to achieve the best possible balance between the storage cost of each option and its suitability given a number of algorithmic requirements. Any finite representation of directional functions is based on the selection of a number of basis functions. The representation of a distribution then consists of its coordinate vector in the chosen basis.

Previous algorithms employ for example constant basis functions defined over the cells of a "global cube" [4], or spherical harmonics basis functions up to a prescribed order [2, 7]. The global cube approach has the advantage of simplicity, first because it is very easy to manipulate, but also because function products can be evaluated easily (since the basis functions have non-overlapping support). However it is inherently a discontinuous representation, prone to disturbing rendering artifacts.

Spherical harmonics, on the other hand, always produce continuous functions. But they are non-zero over the entire hemisphere, making the computation of function products much more expensive.

4.1 Spherical Harmonics

In our implementation we use spherical harmonics basis functions. These form an orthogonal basis of the set of distributions on the unit sphere. This infinite collection of basis functions is typically denoted by $Y_{l,m}(\theta, \phi)$ where $0 \leq l < \infty$ and $-l \leq m \leq l$. In direct analogy with a Fourier series in one dimension, any square-integrable function, $f(\theta, \phi)$, can be expressed in this basis, with a set of scalar coefficients $C_{l,m}$.

An approximate representation of a directional function is obtained by storing only the first few coefficients of this decomposition, up to a given maximum level. BRDFs can be encoded by such vectors of coefficients for use in a radiosity simulation [7].

Representation of diffuse surfaces using Tangent-Sphere functions In the diffuse case, all radiant intensity distributions are combinations of oriented Tangent-sphere functions (see Equation 13).

The decomposition of $T_{\vec{n}}(\vec{u})$ into spherical harmonics can be computed for a given direction $\vec{n}$. The simple shape of this function allows a very good approximation with only 9 coefficients ($l \leq 2$). The coefficients of this decomposition are thus functions of $\vec{n}$, and they can themselves be decomposed using spherical harmonics of $\vec{n}$. This double decomposition was already used by Westin *et al.* to represent anisotropic BRDFs [12]. In our case it is stored in a data file, since the Tangent-sphere function is always the same.

The spherical harmonics representation of $T_{\vec{n}}(\vec{u})$ is obtained by evaluating the value of each coefficient for the direction $\vec{n}$. Since this only depends on the surface orientation, it is only performed once in the program, and is then stored with the polygon (and thus shared by all hierarchical elements on the surface).

Computation of the scattering integral If incident radiance is stored with the clusters, the integral in Equation 12 must be evaluated at each cluster-surface interface. The convolution of incident radiance and the BRDF is quite costly to compute, especially since function products are difficult to express with spherical harmonics coefficients. We are currently investigating an efficient algorithm to compute such convolutions, based on the use of recurrence relations, and the observation that the integral of a function is represented by it's ($l = 0$,$m = 0$) coefficient.

5 Implementation and First Results

We have implemented the representation of radiant intensity and the equivalent push/pull operation in our testbed clustering system. As described above we have used spherical harmonics for the representation of directional functions. Our implementation is still preliminary in the sense that for now a limited number of orientations are allowed for non-diffuse surfaces. The color plates in the appendix demonstrate the versatility and high potential of the method.

5.1 Directional properties for clusters of diffuse surfaces

We first consider the anisotropic behaviour of clusters containing only diffuse surfaces. Plate 2 shows an example with over 6,000 surfaces. The ceiling receives no primary illumination, and is only illuminated by light reflected by the cluster. We see that the pattern of light on the ceiling is displaced with respect to the vertical direction.

As an indication to the reader of the relative cost of the storage of directional radiant intensity, comparisons are made to images generated using the algorithms presented in [6, 8], in which directional functions are not used. Since the refinement criteria are no longer the same, we set our subdivision threshold so that the two executions result in similar number of links refined for two iterations. The following table gives the computation time (in seconds) and memory cost (in Mb) for directional (dir) and traditional (trad) clustering algorithms.

Name	Polygons	Time (dir)	Time (trad)	Mem. (dir)	Mem. (trad)
Simple	13	29.6	24.0	8.5	4.7
Cubes	6000	140.0	46.1	13.4	6.9

We see that the computation time for the directional approach is between 20% to 3 times higher. This can be explained by the additional expense in combining the tangent sphere functions and the directional representations of radiant intensity. The comparisons are given only as an indication; in the resulting images for the $Cubes$ scene the directional algorithm obtains a much higher quality representation of the secondary illumination on the ceiling (see Plate 2).

The memory requirements for the directional representation are approximately twice that of the traditional clustering approach. These numbers are more meaningful since they are not affected as much by the different refinement criteria. If the growth factor is close to the indicated factor of two, this implies that memory utilization does not pose a major problem for our approach, since even very complex scenes will not require unmanageable amounts of memory.

5.2 Results for general reflectors

Plates 3 and 4 show simulations performed with a cluster of glossy surfaces. Both directional reflection and directional attenuation are demonstrated, by illuminating the scene from two different directions. Plate 5 illustrates the view-dependent character of radiant intensity distributions, with two different views of the same scene. Computation times for all these images range from 17 to 103 seconds.

6 Discussion and Conclusions

We have presented a general framework for the hierarchical representation of energy exchanges taking into account the non-uniform directional behavior of surfaces and object clusters. Although conceptually simple, this approach raises a number of practical issues, which we discuss below.

Benefits and limitations of the Approach The explicit representation of directional radiant functions for object clusters has several important benefits. First, it allows a smooth integration of non-diffuse reflectors in a clustering algorithm. Second, if incoming radiance is stored explicitly, it reduces the asymptotic complexity of the clustering algorithm. Third, the consideration of directional extinction properties greatly improves the applicability of the approximate transmission calculation based on the volume analogy. Finally, the method allows the simulation of non-isotropic scattering volumes with arbitrary phase functions. In practice we consider that the most useful feature is the ability to mix diffuse and non-diffuse reflectors in a scene at a moderate additional cost. In particular the overhead costs for diffuse reflectors remain reasonable, while allowing much more accurate transfers between clusters. We tend to prefer the option of implicit storage for incident radiance, since it appears very difficult to do away completely with any traversal of the hierarchy during the gathering stage. For instance, the consideration of intra-cluster visibility is much easier when each contribution is pushed down to the surfaces [8]. The efficient representation of directional functions is a difficult issue. For general reflectors many spherical harmonics coefficients may be needed, resulting in high storage and computation costs.

Future directions A major area of research for future work is the investigation of multi-resolution representations of directional functions. It may be possible to store different levels of detail at each cluster, instead of storing a complete distribution everywhere. This would dramatically lower the storage costs, while allowing true multi-resolution visibility computation through object clusters [8]. Another interesting direction is the computation (and storage) of complete scattering functions for all clusters. These will allow the direct transformation of incoming radiance to radiant intensity, similar to a volumic phase function. However the storage costs for such bidirectional phase functions may be prohibitive.

References

1. Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '93 (Anaheim, CA, USA), pages 155–162. ACM SIGGRAPH, New York, August 1993.
2. Brian Cabral, Nelson L. Max, and Rebecca Springmayer. Bidirectional reflection functions from surface bump maps. *Computer Graphics*, 21(4):273–281, July 1987. Proceedings SIGGRAPH '87 in Anaheim (USA).
3. Pat Hanrahan, David Saltzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, August 1991. Proceedings SIGGRAPH '91 in Las Vegas (USA).
4. David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics*, 20(4):133–142, August 1986. Proceedings SIGGRAPH '86 in Dallas (USA).
5. Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. In *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '94 (Orlando, FL). ACM SIGGRAPH, New York, July 1994.
6. François Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *to appear in IEEE Transactions on Visualization and Computer Graphics*, 1(3), September 1995. (a preliminary version appeared in the fifth Eurographics workshop on rendering, Darmstadt, Germany, June 1994).
7. François Sillion, James Arvo, Stephen Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics*, 25(4):187–196, August 1991. Proceedings SIGGRAPH '91 in Las Vegas (USA).
8. François Sillion and George Drettakis. Feature-based control of visibility error: A multiresolution clustering algorithm for global illumination. In *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '95 (Los Angeles, CA). ACM SIGGRAPH, New York, August 1995.
9. François Sillion and Claude Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics*, 23(3):335–344, August 1989. Proceedings SIGGRAPH '89 in Boston (USA).
10. Brian Smits, James Arvo, and Donald P. Greenberg. A clustering algorithm for radiosity in complex environments. In *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '94 (Orlando, FL). ACM SIGGRAPH, New York, July 1994.
11. John R. Wallace, Kells A. Elmquist, and Eric A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics*, 23(3):315–324, July 1989. Proceedings SIGGRAPH '89 in Boston.
12. Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. *Computer Graphics*, 26(4):255–264, July 1992. Proceedings of SIGGRAPH '92 in Chicago (USA).

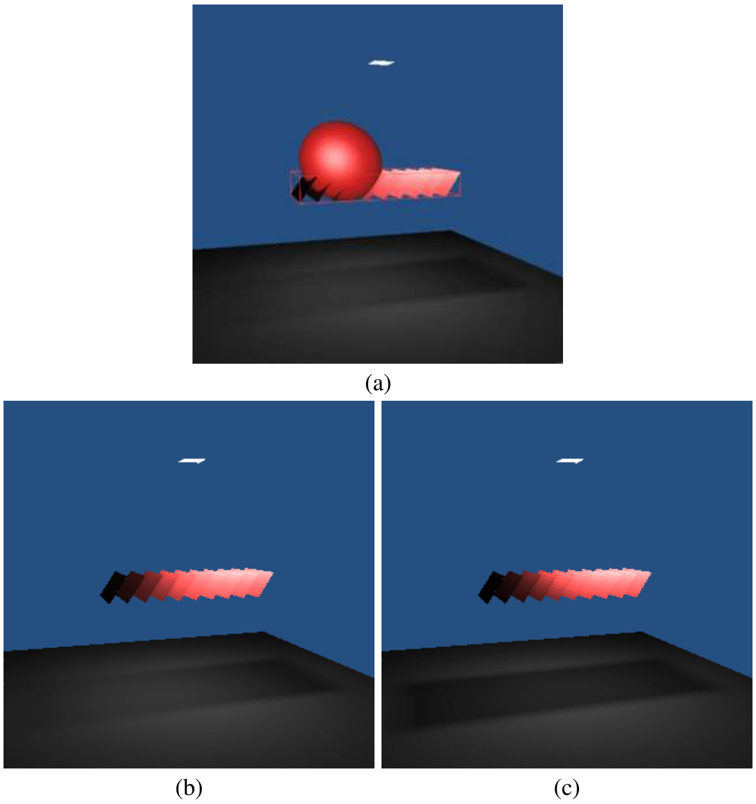

Plate 1. Using directional visibility information: (a) representation of the directional extinction coefficient for the cluster of slanted objects. (b) Simulation showing the varying attenuation in the shadow area. (c) Simulation using isotropic extinction: note the uniform attenuation in the shadow area.

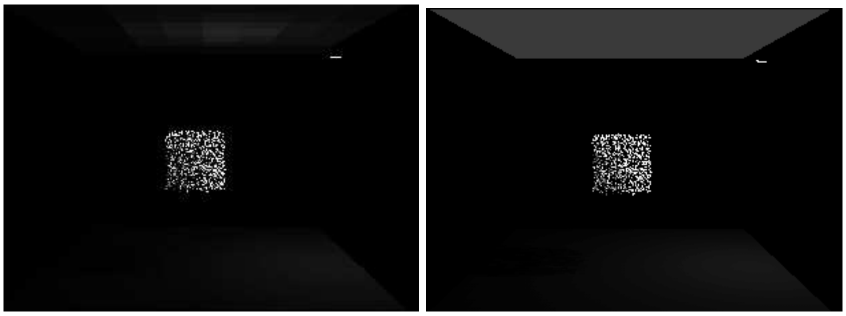

Plate 2. Solution for a scene with 6000 diffuse surfaces. (a) directional and (b) non-directional clustering.

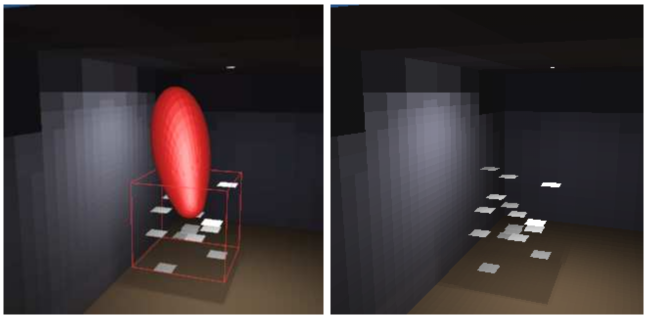

Plate 3. Simulation with a cluster of specular reflectors (overhead illumination): (a) distribution of radiant intensity for the selected cluster. (b) final image.

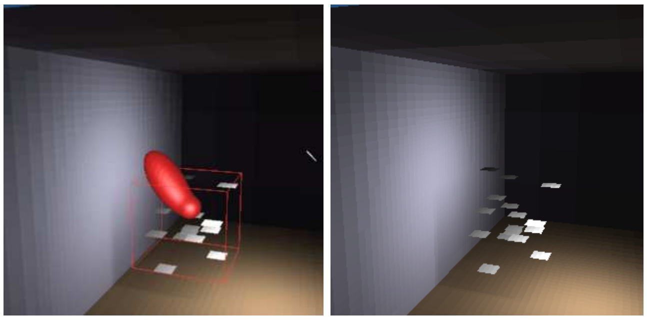

Plate 4. Simulation of the same scene (with illumination coming from the side). Comparing to Plate 3, note the change in secondary illumination and the change in the cluster's shadow.

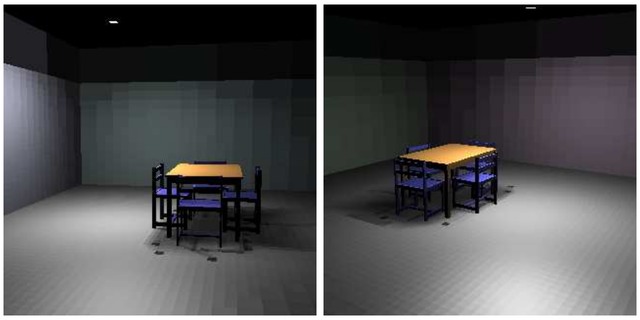

Plate 5. Two views of a scene with glossy surfaces (floor and table top). Note the differences in the appearance of the non-diffuse surfaces.