

A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters

François X. Sillion

Abstract— This paper presents a new radiosity algorithm that allows the simultaneous computation of energy exchanges between surface elements, scattering volume distributions, and groups of surfaces, or object clusters. The new technique is based on a hierarchical formulation of the zonal method, and efficiently integrates volumes and surfaces. In particular no initial linking stage is needed, even for inhomogeneous volumes, thanks to the construction of a global spatial hierarchy. An analogy between object clusters and scattering volumes results in a powerful clustering radiosity algorithm, with no initial linking between surfaces and fast computation of average visibility information through a cluster. We show that the accurate distribution of the energy emitted or received at the cluster level can produce even better results than isotropic clustering at a marginal cost. The resulting algorithm is fast and, more importantly, truly progressive as it allows the quick calculation of approximate solutions with a smooth convergence towards very accurate simulations.

I. INTRODUCTION

The recent introduction of hierarchical formulations of the radiosity method has been a major step in allowing rapid simulation of energy exchanges between surfaces. In this paper, we first demonstrate that a similar formulation is possible for volume transfers, and propose a unified algorithm that transparently deals with scattering volumes and surfaces. We recognize that inhomogeneous volumes can be inserted in a global hierarchy, which dramatically reduces the computational cost of the simulation by removing the initial linking stage. We then suggest that the same operation can be performed for groups of surfaces, or *clusters*. The resulting radiosity algorithm is orders of magnitude faster than traditional hierarchical radiosity and efficiently simulates volume and surface scattering.

Previous work

Hierarchical algorithms attempt to limit the amount of subdivision of surfaces by selecting the proper level of detail needed to represent each energy transfer within a given error tolerance [12]. The exchange of energy between a pair of surfaces is represented using a hierarchy of *links* that connect some of their components. Links are created in an adaptive refinement procedure that recursively attempts to link two sub-surfaces. The error incurred when representing the energy transfer using a single link is estimated (a

variety of error measures are available to choose from) and the link is actually established if that error falls below some threshold. Otherwise one of the two sub-surfaces is subdivided and the procedure is called recursively for each of the new components.

The benefit of hierarchical radiosity methods is that the number of links created in the adaptive procedure is linear with respect to the final number of surface elements [11]. This is a major advantage compared to the quadratic matrix of form factors needed to completely represent the possible energy transfers with “conventional” radiosity. However, typical hierarchical radiosity methods proceed in a *top-down* fashion, by selectively subdividing a set of input surfaces. Each pair of input surfaces must still be linked in a preliminary “initial linking” stage, resulting in a quadratic cost with respect to the number of input surfaces. Furthermore, no exchange of energy is computed until the entire “initial linking” phase has terminated.

Clustering

When complex scenes are to be simulated, this quadratic cost rapidly becomes overwhelming, and a bottom-up hierarchical method becomes necessary. One possibility to avoid the quadratic cost is to compute energy exchanges (i.e. establish links) between groups of objects in a single operation. This concept is commonly referred to as “clustering”, since it creates abstract objects that act as representatives of a group (or cluster) of neighboring objects in lighting calculations.

Clustering works by introducing a hierarchy *above* the input-surface level, creating a set of abstract entities that exchange energy. Previous efforts to simplify the radiosity algorithm have either attempted to treat subsets of the scene independently in a divide-and-conquer approach [37], [1], or manually constructed groups of patches used as representatives for their contents [24], [16]. Current research efforts focus on the combination of clustering and hierarchical radiosity [29], [34] and automatic construction of clusters. In particular, independent work by Smits *et al* [34] suggested a clustering algorithm very similar to the one presented below. In this paper we propose a hierarchical algorithm that borrows ideas from volume scattering methods to express the hierarchical exchanges between clusters, and completely automates the creation of clusters. The originality of the present work comes from realizing that since clusters contain several surfaces of arbitrary orientation, they can be represented as volumetric objects. The

F. Sillion works for the “Centre National de la Recherche Scientifique” (CNRS) in the iMAGIS laboratory, a joint research project of CNRS/INRIA/INPG/UJF. Postal address: iMAGIS/IMAG, B.P. 53, F-38041 Grenoble Cedex 9, France. E-mail: Francois.Sillion@imag.fr. Web address: <http://safran.imag.fr>.

insight gained from this analogy results for instance in a novel approach to visibility calculation using equivalent extinction properties. In addition, the algorithm proposed below handles transparently all possible transfers between volume elements, surface elements, and object clusters.

Paper overview

The paper is organized as follows. First we show in Section II how the zonal method – for the computation of energy exchanges between homogeneous isotropic volumes and diffuse surfaces – can be reformulated as a hierarchical algorithm. Section III explains how inhomogeneous objects should be treated in a hierarchical algorithm and introduces the notion of a complete hierarchy to avoid initial linking. In Section IV we propose to represent surface clusters as scattering volumes, allowing the complete elimination of the initial linking stage. Section V discusses the difficulties associated with volume representation and suggests solutions. The specific issues encountered in estimating the volumetric properties of clusters are reviewed and solutions are suggested.

II. HIERARCHICAL FORMULATION OF THE ZONAL METHOD

In this section we present a hierarchical formulation for the simulation of energy exchanges between isotropic scattering volumes and diffuse surfaces. We begin by presenting the governing equations for light transfer with participating media, and continue with a brief account of the traditional zonal method [25]. We then explain how to combine the zonal method and a hierarchical radiosity algorithm [12]. A similar combination, limited to homogeneous media, has been described elsewhere [5]. The algorithm described here is more general in that it accommodates arbitrary density distributions as shown in Section III.

A. Energy transfer in volume densities

The presence of a participating medium can have a number of effects on the distribution of light, as evidenced by the appearance of fog, haze, smoke and dust. Radiance can be reduced by absorption or scattering and enhanced by emission or scattering from other directions. Thus scattering can account for both a decrease or an increase of radiance. The terms *out-scattering* and *in-scattering* are used to describe the scattering of light away from a given direction, or into a given direction, respectively, as shown in Fig. 1.

A.1 A general transfer equation

In the presence of a participating medium, light can interact with matter at any point in space. Thus radiance must be defined everywhere, and a general integro-differential equation is used to express the local energy balance [14]. This equation expresses the rate of change in radiance along a path as:

$$\frac{dL}{ds} = -\kappa_t L + \kappa_a L_e + \kappa_s \int_{\Omega} L_i(\Theta) f(\Theta) d\omega, \quad (1)$$

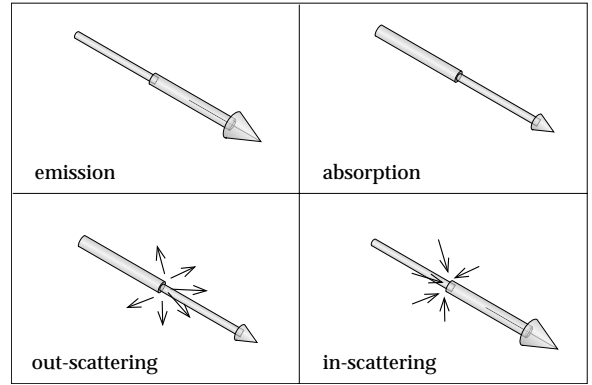


Fig. 1. Processes contributing to the variation of radiance along a path at a particular differential volume element.

where s is the path variable (a measure of the distance traveled along the path from the origin), and Ω denotes the entire sphere of directions [15]. The various terms of this equation correspond to the following physical phenomena:

- $-\kappa_t L$ represents the attenuation of radiance due to absorption or out-scattering. κ_t is called the *extinction coefficient* and is defined as

$$\kappa_t = \kappa_a + \kappa_s.$$

The properties of the medium are represented by the two scalar functions κ_a (called the *absorption coefficient*) measuring the fraction by which radiance is reduced per unit length due to absorption, and κ_s (the *scattering coefficient*) measuring the fraction by which radiance is reduced per unit length due to out-scattering. Both quantities have dimensions of inverse length.

- $\kappa_a L_e$ represents the increase in radiance due to emission by the medium. L_e is the emitted radiance, and the presence of κ_a is a consequence of reciprocity principles from thermodynamics [28].
- The integral term represents the increase in radiance due to in-scattering, that is the effect of light coming from all directions and scattered into the direction of interest. $f(\Theta)$ is the *phase function*, describing the directional probability distribution for scattering from direction Θ into the direction of the path.

A.2 Transmittance

Consider a pencil of light traveling from a surface into a participating medium (see Fig. 2). Only a fraction of the light leaving the surface will pass through the medium without being absorbed and scattered, and this fraction (called the *transmittance*) can be computed by considering the extinction term alone in Equation 1:

$$\frac{dL}{ds} = -\kappa_t L. \quad (2)$$

This simplified equation can be integrated to yield the radiance at a distance s into the medium

$$L(s) = L(0) e^{-\int_0^s \kappa_t(u) du}. \quad (3)$$

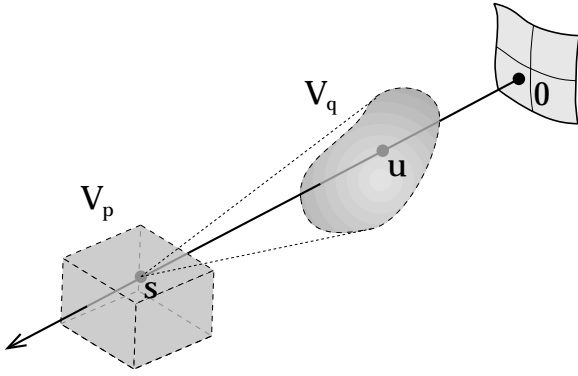


Fig. 2. Geometry for the energy transfer equations in a participating medium.

Therefore the transmittance of the medium along the path is given by:

$$\tau(s) = e^{-\int_0^s \kappa_t(u) du}. \quad (4)$$

B. Principles of the zonal method

One difficulty in the formulation of the problem using Equation 1 is that radiance is both *produced* locally in the medium as a result of emission and in-scattering, and *transmitted* through it. Thus it is not a local quantity, which precludes the use of most solution techniques based on finite elements.

B.1 Formal solution using source radiance

A formal solution of the general transfer equation is obtained by introducing a new quantity to describe the local production of light, and expressing the relationship between radiance and this quantity. Equation 1 can be rewritten as

$$\frac{dL}{ds} = -\kappa_t L + \kappa_t J, \quad (5)$$

where the *source radiance* J is defined by:

$$J(s) = (1 - R(s)) L_e + R(s) \int_{\Omega} L_i(\Theta) f(\Theta) d\omega. \quad (6)$$

This quantity accounts for both emission and in-scattering at s . $R(s)$ is the *scattering albedo* of the medium, expressing the fraction of attenuation due to out-scattering:

$$R(s) = \frac{\kappa_s}{\kappa_t}.$$

A formal solution of Equation 5 is

$$L(s) = L(0)\tau(s) + \int_0^s J(u)\tau(s-u)\kappa_t(u)du. \quad (7)$$

Equation 7 can be used as the basis of a simple ray tracing algorithm, since the radiance leaving the medium in the direction of the eye is obtained by a simple one-dimensional integral. This derivation also provides a justification for the

simple fog models commonly found in rendering packages (and in special-purpose hardware accelerators) and for volume rendering algorithms [7].

This formal manipulation suggests that if the source radiance J can be computed in a first pass, the final radiance can then be obtained using for example a ray tracing pass based on Equation 7.

B.2 The zonal method for isotropically scattering media

The issue of computing the source radiance J is somewhat simpler than the general transfer problem because of its local character. Still it is very difficult to solve in the general case of an anisotropic scattering medium with arbitrary phase functions. The *zonal method* solves for J in the simplified case of an isotropically scattering medium: that is, a medium with a constant phase function

$$f_{\text{isotropic}} \equiv \frac{1}{4\pi}.$$

This is done in a typical finite-element way, similar to the traditional radiosity method [33], by first breaking up the volume containing the participating medium into discrete elements, used in addition to the usual surface patches. Radiosities are defined for both surfaces (with $B_i = \pi L_i$) and volumes (with $B_k = \pi J_k$). The source radiance J does not depend on direction since the method assumes that the medium has isotropic scattering and emission.

Using these notations, the total power emitted by an element is given by [25]

$$P_i = A_i B_i \quad \text{for a surface element} \quad (8)$$

$$P_k = 4\kappa_k V_k B_k \quad \text{for a volume element} \quad (9)$$

In these equations, A_i is the area of the corresponding surface patch, V_k is the volume of the volume element, and κ_k is the average extinction coefficient for that volume element. Thus the quantity $4\kappa V$ can be seen as the “equivalent area” of a volume element¹.

As in the traditional radiosity method, the zonal method proceeds by setting up a linear system of equations to express the coupling between all variables based on Equation 1 [25], [23]:

$$B_p = E_p + \rho_p \sum_q F_{pq} B_q. \quad (10)$$

Here the summation is carried out both over a set of surface patches, and a set of volume elements, each of which is assumed to possess a uniform radiosity value. ρ denotes either the diffuse reflectance (for a surface) or the albedo (for a volume). The expression of the form factor F_{pq} depends on the physical nature of elements p and q .

B.2.a Volume to volume form factor. As an example we show how to derive the form factor between two volume

¹In particular it has units of $m^{-1}m^3 = m^2$.

TABLE I
EXPRESSION OF THE FORM FACTORS FOR ALL COMBINATIONS OF SURFACES AND VOLUMES.

$F_{\text{row-column}}$	Surface j	Volume k
Surface i	$\frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\tau \cos \theta_i \cos \theta_j}{\pi r^2} dA_j dA_i$	$\frac{1}{A_i} \int_{A_i} \int_{V_k} \frac{\tau \kappa_k \cos \theta_i}{\pi r^2} dV_k dA_i$
Volume m	$\frac{1}{V_m} \int_{V_m} \int_{A_j} \frac{\tau \cos \theta_j}{4\pi r^2} dA_j dV_m$	$\frac{1}{V_m} \int_{V_m} \int_{V_k} \frac{\tau \kappa_k}{4\pi r^2} dV_k dV_m$

elements. Consider the radiosity at point s given by Equation 6:

$$B(s) = \pi J(s) = \pi \left[(1 - R(s)) L_e + \frac{R(s)}{4\pi} \int_{\Omega} L_i(\Theta) d\omega \right]. \quad (11)$$

Using Equation 7 to express the incident radiance, we have

$$B(s) = (1 - R(s)) B_e + \frac{R(s)}{4} \int_{\Omega} \int_0^s J(u) \tau(s-u) \kappa_t(u) du d\omega. \quad (12)$$

Consider the contribution to $B(s)$ of a given homogeneous volume V_q (Fig. 2). Computing the integrals in Equation 12 over V_q and changing the integration variable ($dV = r^2 du d\omega$) we obtain

$$B_{\leftarrow q}(s) = R(s) B_q \int_{V_q} \frac{\tau(s-u) \kappa_t(u)}{4\pi r^2} dV. \quad (13)$$

The form factor between V_q and a volume V_p is obtained by estimating the total power received by V_p (integrating the quantity $4\kappa_t(s) B_{\leftarrow q}(s) dV_p$) and dividing by its equivalent area:

$$F_{pq} = \frac{1}{V_p} \int_{V_p} \int_{V_q} \frac{\tau \kappa_t}{4\pi r^2} dV_p dV_q.$$

B.2.b Other form factors. The four possible cases for the form factors are summarized in table I: In this table τ is the transmittance between differential elements, A_i is the area of patch i and V_k is the volume of volume k (Fig. 3). This presentation differs from the original zonal method in that it favors the simplicity of the transfer equation over that of the form factor and the reciprocity relations. The resulting equation is identical to the surface radiosity equation [33] and allows a very smooth integration of surfaces and volumes in a unified algorithm.

C. Hierarchical zonal algorithm

In this section we show that the hierarchical radiosity algorithm is easily adapted to a mixture of surfaces and volumes. We assume that the scene is described as a set of surfaces and a set of volume elements containing participating media. We further assume that a hierarchical subdivision mechanism is available for both types of objects. In the case of surfaces, a quadtree in parameter space is often

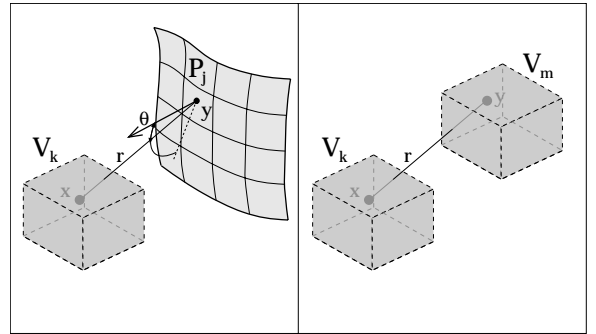


Fig. 3. Geometry for the computation of exchange factors. (a) surface-to-volume factors. (b) volume-to-volume factors.

used to represent this subdivision. For volumes a variety of structures are possible, including octrees and binary trees. In our implementation we have used a tri-dimensional $k-d$ tree [4], [27].

In the remainder of this paper we will refer to portions of surfaces and volumes as generic *Hierarchical Elements* or H-elements. We now show that the hierarchical radiosity algorithm of Hanrahan *et al.* applies easily to such H-elements, and that most operations are generic in the sense that they do not need to know whether they apply to volumes or surfaces. In particular this is the case with the hierarchical refinement, gathering and Push-Pull procedures.

C.1 Hierarchical radiosity for H-elements

Let us review the basic operations of a hierarchical radiosity procedure as proposed by Hanrahan *et al.* [12].

C.1.a Hierarchical refinement. The core of the hierarchical radiosity algorithm is a recursive refinement procedure: considering two H-elements, it decides whether the transfer of energy between them can be correctly represented. If yes, it establishes a *link* between them and computes their form factor (this computation is explained in Section II-C.2), otherwise it subdivides one of the elements and recurses.

The decision to subdivide is typically based on a computed bound on the error incurred by a link at the current level. This can be done either by bounding the form factor alone[11], by bounding the radiosity transfer (BF refine-

ment) [12], or by estimating an error bound on the energy transfer [18]. Importance weighting of these bounds is also possible [35]. Our current implementation uses BF refinement.

Thus the simulation procedure starts with an *initial linking* phase, where the potential interaction is represented for each pair of objects, by a link between their top-level H-elements. The recursive refinement procedure is then applied to all initial links.

C.1.b Radiosity Solution. Once links have been refined, energy transfers are computed by gathering radiosity at all elements. For each object, the hierarchy of H-elements is traversed, and for each of these H-elements, radiosity is gathered across all its links. The gathering operation is equivalent to a step of a Gauss-Seidel iterative solution of Equation 10.

C.1.c Push - Pull. After each gathering operation, contributions received at different hierarchical levels of a given object must be combined to ensure that each H-element possesses an accurate view of its total radiosity. This is accomplished by a bidirectional traversal of the hierarchical structure (Push-Pull procedure) that pushes radiosity received at higher levels down to their children, adding it to the children’s radiosity, then pulls the children’s radiosity up the hierarchy, averaging them along the way.

A converged radiosity solution is obtained, for the current set of links, by iterating the combined gather and Push - Pull operations until no H-element experiences a significant change in radiosity.

Note that all of the above operations are performed in a uniform manner for both types of H-elements, that is, surfaces or volumes.

C.2 Form factor computation using generic operations

There are four possible cases for the form factor between a pair of H-elements, since each can be either a volume or surface element. However using an object-oriented approach it is easy to define generic building blocks that behave differently for different types of hierarchical elements, and allow easy computation of all form factors.

Thus we express the four form factor formulas given in Table I using the generic formula:

$$F_{pq} = \mathcal{S}_p \mathcal{S}_q \frac{1}{\int_p d\mathcal{A}_p} \int_p \int_q \frac{\tau \mathcal{R}_p \mathcal{E}_q}{\pi r^2} d\mathcal{A}_p d\mathcal{A}_q \quad (14)$$

where the following functions are defined differently for volumes and surfaces (this is easily accomplished using for instance the C++ overloading mechanism).

name	symbol	surface	volume
Receiver factor	\mathcal{R}_p	$\cos \theta$	1
Emitter factor	\mathcal{E}_p	$\cos \theta'$	1
Area factor	\mathcal{A}_p	A	$4\kappa V$
Scale factor	\mathcal{S}_p	1	$\frac{1}{4}$

In our implementation we provide generic access to sampling routines that generate quadrature points across each

H-element. The form factor between any two H-elements can then be computed without knowing their type.

Note that the transmittance factor in the form factor expressions is the “equivalent” of the visibility term in traditional surface form factors. It can be ignored in the computation of an upper bound of the form factor, but must be evaluated for the final form factor estimate.

C.3 Image Display

Recall that an image can be generated from a solution of Equation 10 using the line integral from Equation 7. This can be done using any volume rendering package. A ray-casting based technique is likely to provide the best results in general. In our implementation we have chosen to use a direct projection (“splatting”) algorithm [36], [17] in order to maintain the short rendering times necessary for interactive simulation: Surfaces are rendered first using a conventional z-buffer. The leaf H-elements of all volumes are then composited in back-to-front order using a semi-transparent polygonal representation. Note that the use of a global *k-d* tree or octree, as suggested in the next section, allows easy back-to-front ordering of all volume H-elements. In the absence of such a global structure, it may be difficult to properly render overlapping volumes.

III. A COMPLETE HIERARCHICAL ALGORITHM FOR INHOMOGENEOUS OBJECTS

Section II showed that a hierarchical radiosity system can easily be adapted to accommodate participating media. In this section we examine two implications of considering volumes, that require subtle modifications of the hierarchical radiosity algorithm. First we consider what happens when an object (surface or volume) possesses inhomogeneous radiometric properties. We then introduce self-links to represent energy exchanges inside a volume, and show that the cost of initial linking can be eliminated for volumes by using a global hierarchy. Finally we demonstrate some results of the combined surface-volume radiosity algorithm.

A. Hierarchical representation of energy exchanges for inhomogeneous objects

A typical assumption of radiosity algorithms is that a surface patch has a uniform reflectance. The Push-Pull procedure described above relies on this assumption, since it propagates radiosity received by a patch at a given level of the hierarchy in a uniform manner to all its children.

The introduction of participating volumes stresses the unnecessary limitation imposed by the homogeneous assumption: in many cases participating media are described by a varying density or albedo distribution. A volume with inhomogeneous density or albedo is in fact the radiative “equivalent” of a textured surface: in both cases the hierarchical distribution of energy exchanges must take into account the varying optical properties.

Consider an H-element that has been linked to a light source. The link represents the fact that some energy is received at that level. However if the H-element is not homogeneous in terms of reflectance/albedo, some areas

will reflect more of this energy than others. Thus radiosity can not be simply pushed down the hierarchy. Using the terminology of Arvo *et al.* [2], the *local reflection operator* must consider the inhomogeneous reflective character of the surface.

Fortunately, this can be accomplished quite simply by using the underlying hierarchy to apply this local reflection operator. In effect all the necessary information is propagated down the hierarchy to allow the application of the local reflection operator only at *leaf* H-elements. In this manner the correct balance of energy is always maintained. Density and albedo informations are not used in the same way, thus while albedo information is only used at leaves of the hierarchy, density values are volume-averaged throughout the structure. This ensures that a proper “equivalent area” can be used for all H-elements.

In practice, the gathering procedure is modified so that *Irradiance*, or incoming energy per unit area, is gathered across a link. The irradiance received by H-element p across a link to H-element q is $I_{p \leftarrow q} = F_{pq} B_q$. The Push-Pull procedure is also modified to push irradiance down the hierarchy, transform irradiance into radiosity at leaf H-elements, then pull radiosity up the hierarchy. A similar algorithm was described by Gershbein *et al.* for wavelet-based radiosity [9].

Note that in the case of volume elements, the “area”-averaging performed when pulling up radiosity values uses the “area factor” described above, thereby ensuring proper distribution of energy for volumes of inhomogeneous density. The gathering procedure is described with pseudocode in Appendix A.

B. Self linking

In surface-based radiosity, surfaces are generally not allowed to interact with themselves². Thus no form factor is computed from a surface to itself, and no link is ever established from a patch to itself. For volumes the situation is different, and the volume-volume form factor F_{mk} in Table I is non-zero when $m = k$. Therefore in general there must be a link from a volume to itself (called here a *self-link*), that represents all the interactions taking place *inside* the volume.

The subdivision of a self-link is slightly different from that of a link between two different elements: In this case links must be created between all possible pairs of hierarchical children, thus including a self-link for each child, as illustrated in Fig. 4.

By combining the improved treatment of inhomogeneities and the notion of self-links it is possible to eliminate the cost of initially linking all pairs of volumes: the distribution of scattering volumes in the scene can be integrated in a single tridimensional hierarchical structure (using for example a k - d tree or an octree). The entire initial linking phase is replaced by the creation of a single link, from the root of the hierarchy to itself. This unique

²This is a byproduct of the common use of convex surfaces. Nothing in the radiosity equations prevents energy exchanges from a surface to itself, but the form factor is null for convex patches.

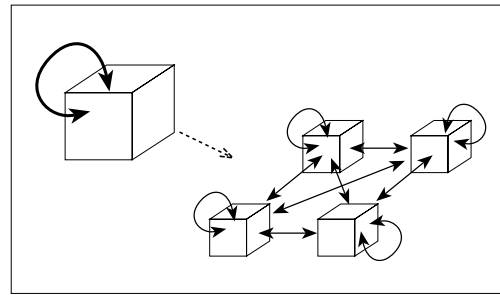


Fig. 4. Subdivision of a self-link, for a volume with four children.

initial link represents all the energy transfers between volumes and is refined by the usual recursive procedure just like any other link.

The elimination of initial linking is the source of a major speedup for the simulation system, since the explicit quadratic dependency in the number of volume elements is eliminated. Initial links between pairs of surfaces, or between surfaces and the root volume, will be eliminated similarly by the clustering mechanisms described in Sections IV and V.

C. Results

The images in Fig. 5 demonstrate the application of the hierarchical radiosity algorithm to volumes and surfaces. A cloud, modeled by a fractal density distribution, is placed in a room with two obstacles, also modeled as volumes. In each row of images the extinction coefficient of the cloud is proportional to its density, with different scaling factors to produce varying scattering conditions. Note that shadowing by the cloud and shadowing of the cloud (both by obstacles and by itself) are automatically simulated thanks to the transmittance calculations.

Different cloud densities produce different strengths for the shadow of the cloud, and dramatically different variations of the cloud’s illumination. Note in particular how the denser medium “captures” more light in its upper layer and acts as a powerful secondary emitter towards the upper parts of the scene. The horizontal obstacle has a transmittance of 25%, and the vertical obstacle is virtually opaque ($\tau \approx 1\%$).

Direct lighting solutions for these images took between 10 to 12 minutes to compute on an SGI Indigo R4000 workstation (all timings were computed on this machine). Three multigridding iterations were run to obtain a complete solution in 20 to 25 minutes. For each solution about 350,000 links were established (out of a total of about 4 billion potential links).

Images were obtained by direct volume rendering (splatting) according to Equation 7.

The image in Fig. 6 was computed using our algorithm with a scene composed of 1,200 polygons and four different density distributions to model the fire and smoke. Computation time was 2 hours and 18 minutes. Note that a volume distribution was used as a light source to model the fire. In the absence of a physical model for the fire, the

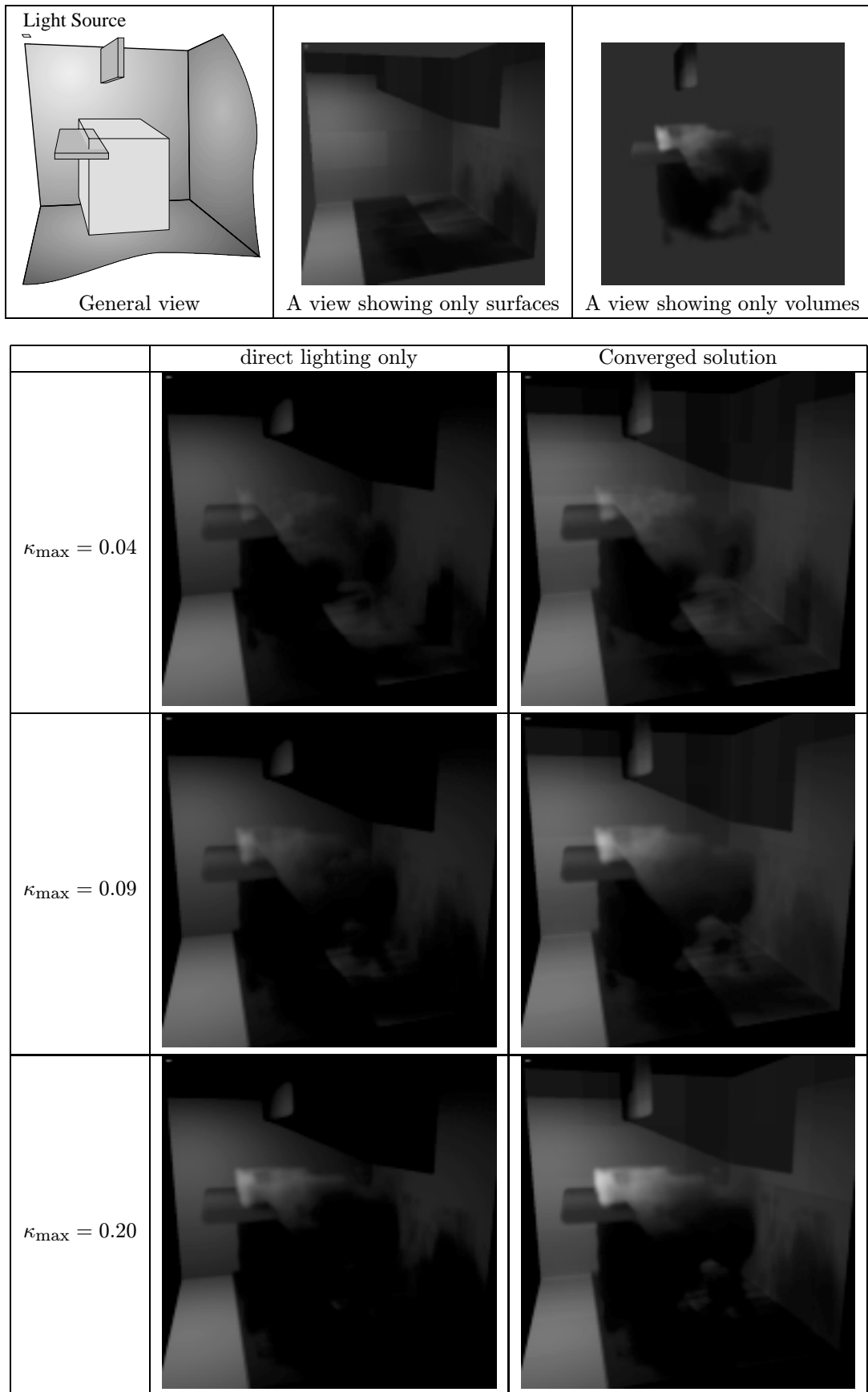


Fig. 5. Simulation of energy transfers in a fictitious density distribution, for different cloud densities. The top three images describe the various components of the scene. In particular, the view of the volumes against a colored background shows the partial transparency effects while the view of the surfaces demonstrates shadows cast on the surfaces by the volume densities.

same cloud model as above was used.

These results show that the hierarchical radiosity algorithm described in this section provides a new tool for the simulation of energy transfer, including multiple scattering, in scenes containing both surfaces and isotropic volume scatterers. This approach complements previous ones based on ray casting or discrete ordinates methods [6], [20].



Fig. 6. Radiosity solution for a scene with volumes and surfaces.

IV. UNIFICATION OF SURFACES AND VOLUMES USING OBJECT CLUSTERS

The analysis of the complexity of hierarchical radiosity algorithms shows that the number of links created is $\mathcal{O}(p + k^2)$, where p is the resulting number of leaf H-elements and k is the number of input objects. Clearly this makes hierarchical radiosity particularly beneficial for scenes where a small number of input objects are subdivided in many elements. However for complex scenes with many small independent objects the complexity gain over standard radiosity techniques is marginal.

Clustering, the operation of grouping together independent objects for the purpose of computing energy exchanges, provides a means to reduce the overall complexity of the simulation [29], [34]. The key issues for clustering are the definition of object clusters, and the calculation of energy transfers between clusters and surfaces.

In this section we propose to build upon the volume scattering ideas presented above and the corresponding hierarchical algorithm: object clusters are treated as volumes that scatter light, and the radiosity of surfaces is obtained from the solution of the volume scattering problem.

A. Creating a volume model

We first consider the case of a large number of “small” objects. This model can be used to represent a number of natural situations such as foliage, as noted by Patmore [22]. The value of this model is that, as objects become smaller, the behaviour of the medium resembles more and more that of a density distribution. In this section we therefore model the radiative properties of surface clusters using equivalent isotropic scattering volumes.

In order to use the above algorithm, a transition from surface clusters to equivalent volumes must be modeled. Rushmeier et al. compute reflectance properties of manually constructed clusters using Monte Carlo simulation [24]. Since we are using a large number of automatically constructed clusters, we need a faster correspondence between surfaces and volume densities, which can be based on the notion of “equivalent area”. As noted above, an isotropic scattering volume element has an equivalent area of $A = 4\kappa V$. Reciprocally, if a number of objects of total surface area A are placed in a volume V , an equivalent extinction coefficient is computed as³

$$\kappa = \frac{A}{4V}. \quad (15)$$

The assumption that objects are “small” allows for a simple construction for the cluster hierarchy: each input surface is inserted in a leaf cell of the 3D hierarchical structure (octree or k - d tree). Equivalent extinction coefficients are then computed using Equation 15 for all leaves, and averaged upwards throughout the hierarchy.

B. Radiosity Solution based on scattering volumes

The equivalence between a collection of small surfaces and a volume density is used in the following sequence of operations. First, a volume density is constructed as explained above. Second, a volume radiosity solution is computed using the hierarchical method of Section II-C. Third, an image is computed with the surfaces shaded according to the volume solution.

We refer to the resulting algorithm as “isotropic clustering”, since object clusters are considered isotropic scatterers. This simple procedure allows two major performance improvements over traditional hierarchical radiosity. Suppression of the initial linking stage results in linear time and space complexity, and a fast approximation to visibility can be computed using extinction properties. However we shall see that the isotropic assumption produces approximate radiosity distributions, adequate for global illumination but possibly insufficient for final display.

B.1 Complexity of isotropic clustering

The cost of establishing a link between two volumes is independent of volume size or density, therefore isotropic clusters can be linked in constant time. Considering the

³Note that in the case of a distribution of infinitely small spheres or cubes, this formula is exact in that it gives the probability of interception of a ray per unit length [13].

expression of the equivalent area of a cluster, and the formula used for its extinction coefficient (Equation 15), the bound computed on the radiosity transfer from an isotropic cluster is equivalent to that of Smits’ “beta-link” [34]. It is simply a coarse upper bound neglecting all orientation and intra-cluster visibility effects.

Each cluster is linked to at most some constant number of other clusters [12], [34], and in the simple scheme described above, the number of clusters is at most linear in the number of input surfaces. Therefore the overall complexity of isotropic clustering is $\mathcal{O}(p + k) = \mathcal{O}(p)$.

B.2 Visibility calculation

The hierarchy of clusters, equipped with their equivalent volume extinction properties, offers an interesting alternative for the estimation of visibility factors.

The transmittance factor in Equation 14 can be understood as a probability of uninterrupted traversal for a ray striking a cluster. Thus the computation of transmittance using extinction coefficients in effect performs a statistical average over the distribution of objects in each cluster. This is interesting whenever an exact visibility computation is not absolutely necessary, since the computation of volume transmittance is cheaper by more than one order of magnitude.

In our implementation transmittance is computed by traversing the hierarchical structure (possibly for a number of sample rays) while accumulating opacity along the way, and then evaluating the negative exponential.

B.3 Obtaining surface radiosity

After the hierarchical radiosity algorithm has finished, volume radiosity is transferred to the surfaces in each leaf cell of the hierarchical structure. Because of the isotropic assumption for the volume, radiosity is simply copied from the volume to the contained surfaces.

C. Results

Fig. 7 shows two different solutions using the isotropic clustering approach, for a test scene containing 6,000 input polygons. The reference image at the top requires over 6 hours of initial linking due to the large number of input polygons. With the clustering strategy described above, equivalent volume scattering properties were assigned to the cells of a spatial subdivision structure, which were then used as clusters in the simulation. Two different error thresholds were selected, resulting in different levels of subdivision. The values of these thresholds correspond to an upper bound on the radiosity transferred over each link. See also Fig. 9 which shows some of the links leaving clusters. The suppression of initial linking results in a dramatic speedup. Also note that the approximate visibility calculation produces fairly good results, for a scene with such a random distribution of objects.

In these images, since the radiance of a volume element is assumed isotropic, it is much lower than the maximum radiance that a single surface would have when subjected

to the same irradiance I : a perfectly reflective diffuse surface perpendicular to the incoming direction would have a radiance of

$$L_{\text{surf}} = \frac{I}{\pi},$$

and the volume element has a radiance of

$$L_{\text{vol}} = \frac{I}{4\pi},$$

thus a difference of a factor of four. In addition, all faces of a cube are shaded according to the same volume, therefore the three-dimensional nature of the objects is not apparent. Both these issues are addressed in the next section.

However the impact on the environment is similar when a large number of cubes is considered. In particular, consider the fact that the entire volume of the cluster participates in secondary transfers, which compensates for the lower radiosity. This suggests that the solution of the volume algorithm is valuable to compute global interreflection effects, although it is not adequate for the direct computation of high-quality images. A “local” shading step can always be added to improve the illumination accuracy on the visible surfaces [19]. However such local computation is typically very expensive, and for many applications defeats the purpose of clustering, that is to quickly obtain approximate solutions while controlling accuracy. The next section shows that it is possible to obtain good quality images without a local pass using a more general clustering strategy.

V. A GENERAL CLUSTERING STRATEGY

The representation of complex groups of surfaces using isotropic volumes is not satisfactory in general. The main reason for this is that the radiance distribution leaving a cluster is generally not uniform. Even when all surfaces are ideally diffuse, reflected radiance can have a strong dependence on direction.

Another important issue with finite-sized surfaces is that when surfaces are not infinitely small, the visibility (transmittance) function becomes more of a binary quantity. Thus the approximation of clusters by semi-transparent media is only valid in the limit of extremely small surfaces. A general clustering algorithm must therefore incorporate a mechanism that computes visibility using either surfaces or volumes, whichever is the most appropriate.

We now describe how to compute energy exchanges between surfaces of arbitrary size using a suitable hierarchy of clusters, and show that the isotropic assumption can be lifted with reasonable cost.

A. Organizing the hierarchy of clusters

We now lift the assumption that all objects are small, and consider the general case of an arbitrary distribution of surfaces and scattering volumes. As shown above, successful clustering requires that a complete hierarchy be constructed, suitable for the recursive refinement procedure of links.

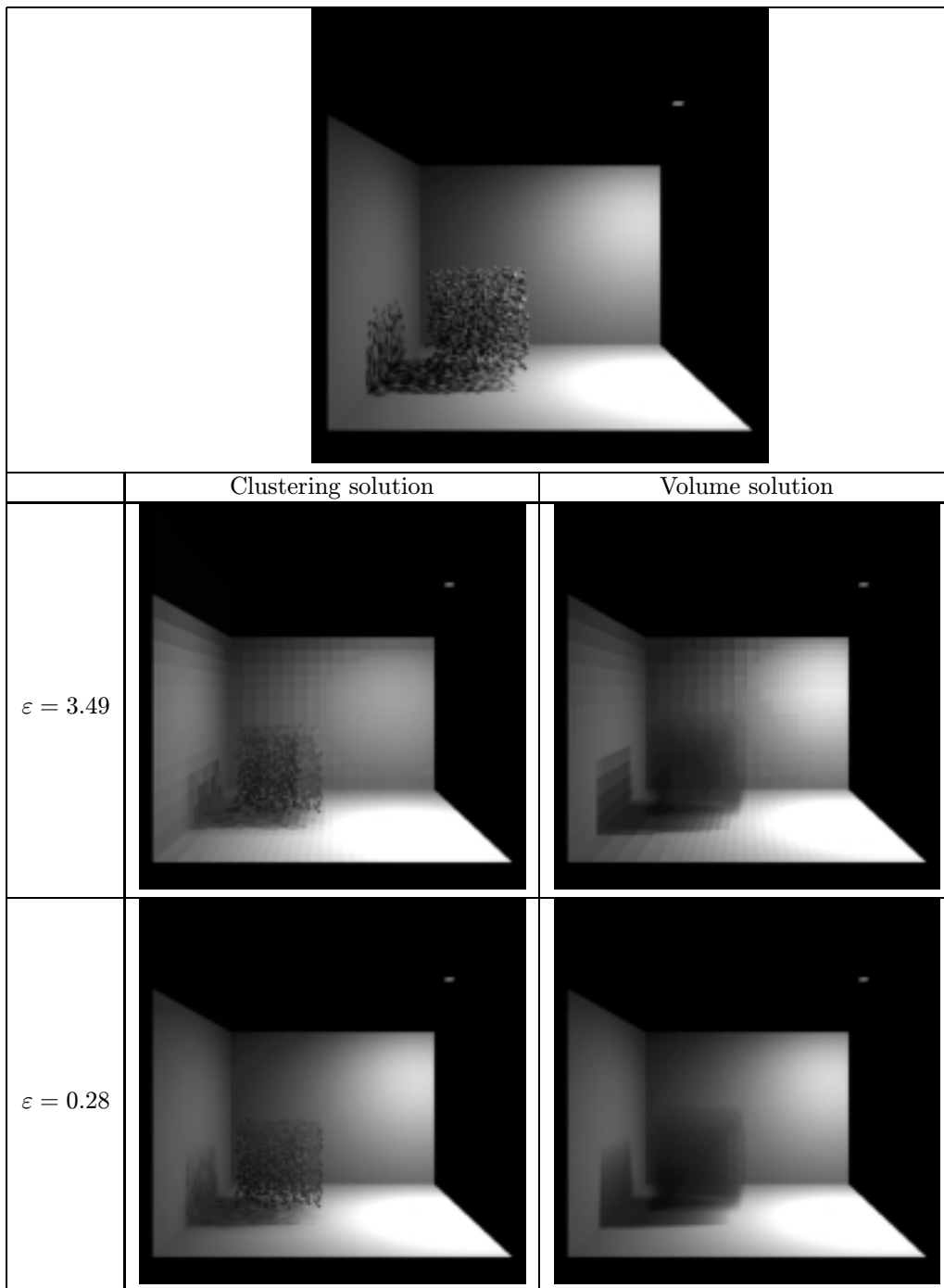


Fig. 7. Isotropic clustering solutions obtained for a test scene with 6,000 input polygons. (top) reference solution, computed with traditional hierarchical radiosity: 6 hours of initial linking, 17 minutes for refinement and solution. (left column) clustering solutions for two different error thresholds: 5.3s and 208s. (right column) equivalent volume solutions, shown to help understand the analogy between clusters and scattering volumes.

To accommodate the most general situations, we construct a hierarchy of H-elements, starting with a root cluster (volume), and obeying the following rules:

- A volume H-element (cluster) may contain a number of surface H-elements. These surface H-elements are considered its children in the hierarchy, in addition to its “normal” volume children, if any.
- Surface H-elements are organized in a classical hierarchy of surfaces.

This general hierarchy is pictured in Fig. 8. When a cluster is subdivided for refinement, it produces a list of children elements that can consist of volume children (if the cluster is not a leaf of the hierarchy) and/or surface children (if surfaces are contained in the cluster). Note that a volume H-element may be a *leaf* from the point of view of the hierarchical volume structure, yet contain surface children (this is also apparent in the pseudo-code of Appendix A). Also, a volume H-element possesses an extinction coefficient, composed in part of the contribution of the contained surfaces, and in part of that of the participating medium, if any.

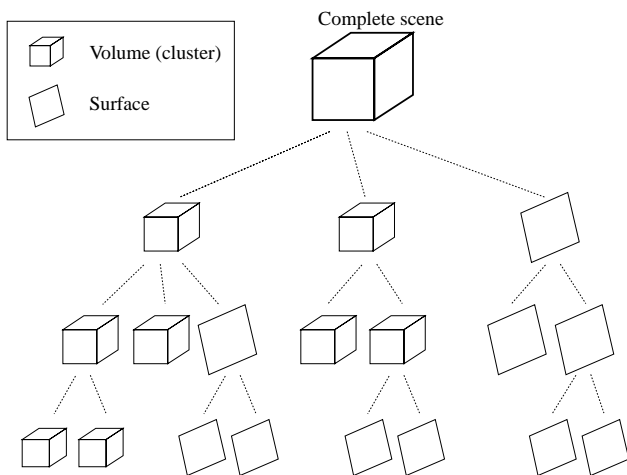


Fig. 8. Hierarchical structure: volumes can have both surface and volume children (left), while surfaces only have surface children (right).

The construction of the cluster hierarchy is a difficult problem, since it has a direct impact on the quality and performance of the simulation. One possibility, similar to the idea of “virtual walls” to accelerate radiosity calculations [37], [8], is to attach surfaces only to leaves of the hierarchical volume description. This can be accomplished for instance by building a hierarchy of bounding volumes around the surfaces [10], [34]. However traditional methods for such a construction do not produce suitable clusters, and current research focuses on their improvement [31].

In the implementation described here, surfaces are attached at the lowest level of the volume hierarchy that contains them entirely. This avoids potential problems when a surface straddles a subdivision boundary, and has additional advantages in terms of visibility testing: by traversing the structure in breadth-first order, potential occluders are tested roughly in order of decreasing size.

B. Hierarchical clustering algorithm

The complete hierarchy of clusters and surfaces is used with the combined surface/volume algorithm of Section II-C. Because the hierarchy encompasses all objects, link refinement proceeds from the root cluster in a top-down manner, and initial linking is completely avoided. In the course of link refinement, arbitrary pairs of surfaces and volumes are considered in a transparent way. The proposed algorithm thus handles surface-cluster, cluster-surface, cluster-cluster and surface-surface energy transfers automatically using procedure overloading.

B.1 Refinement criteria

Fig. 7 showed that the simple estimation of radiosity transfer using isotropic clusters produces approximate results.

One possibility to avoid this difficulty, is to represent directional information at the cluster level. Several data structures have been developed for the simulation of non-diffuse surfaces [30] and could be used to store radiance distributions and scattering functions for each cluster. The benefit of this approach would be to maintain the constant-time linking operation, but the tradeoff with the important cost of the directional structure must be carefully investigated [32].

Smits *et al.* showed that clustering remains attractive even if more work is performed for each link, as long as the total cost of a link is linear in the total number of surfaces in the clusters being considered [34]. Thus a relatively simple way around the problems of isotropic clusters is to allow “smart” energy exchanges between a cluster and its contained surfaces. This is accomplished by modifying the refinement routine, the energy gathering routine and the bidirectional sweep routine (Push-Pull) used to distribute energy in the hierarchy. As mentioned above, the “area”-averaging performed with the modified area factor ensures that the energy is always properly distributed, even when a cluster has a mixture of surfaces and other clusters as its children. The goal is to model the following phenomena in the interaction between two clusters:

- How the radiosity of each surface (and sub-cluster) of the cluster contributes to the energy leaving the cluster.
- How the irradiance on the receiving cluster is distributed among the surfaces and sub-clusters.

As a first approximation, we can ignore all visibility relationships in a cluster and simply take into account the relative orientation of each individual element, with respect to the general direction of transfer. Thus the dot product of each surface’s normal vector and the direction of transfer is used to modulate the irradiance or radiosity (See the pseudo-code in Appendix A). Since the transmittance factor is still only evaluated between the clusters, the additional cost remains reasonable. This is functionally equivalent to Smits *et al.*’s “alpha-link”.

Better criteria can probably be devised, but should remain very simple to implement, to avoid losing all the benefits of clustering. Current work includes the definition of

adaptive criteria taking into account a degraded notion of visibility inside the cluster [31].

B.2 Visibility and transmittance

In our current implementation, transmittance is computed using either volumes or surfaces. Thus the choice is between a fast and approximate method (with volumes and their “equivalent extinction”), and an expensive but more accurate one. A natural extension of this work consists of developing better visibility algorithms that switch automatically from one description to the other. In particular, importance methods [35] should be used to select the appropriate representation for each surface. The hierarchy of clusters also provides the basis for a multi-resolution representation of visibility [31].

C. Results

Fig. 9 shows results obtained with the improved clustering algorithm for the same scene as in Fig. 7. The directional shading of each surface in a given cluster dramatically improves the appearance of the cubes. Careful comparison with the reference solution shows that ignoring all visibility issues inside a cluster produces unwanted illumination of some of the cubes in the lower area. Local visibility can in fact be taken into account in such a way that the total cost of a link remains linear in the number of surfaces in both clusters [31].

The right-hand column shows two sample links from the solutions. As expected, a lower error threshold produces more subdivision, and links are created between smaller clusters. Also note that some of the links connect a cluster to a surface. This is handled transparently by the generic algorithm.

Fig. 10 shows two solutions for a fairly complex scene consisting of about 10,000 input polygons. The radiosity solution is rendered directly using graphics hardware⁴ The combination of clustering and approximate visibility computation allows the calculation of a rough solution in less than 30 seconds, demonstrating that radiosity is a useful tool for preview in design applications. The high-quality solution shows that it is possible to compute fairly accurate images without resorting to a very expensive “local pass”. Note however that the algorithm still depends on the quality of the mesh generated on the surfaces. Light leaks are possible whenever mesh boundaries are not properly positioned with respect to the radiosity function [3]. Using importance weighting with clustering will help directing the computational effort to visible areas, while other areas can be treated as large clusters.

Fig. 11 shows some sample links for the solution of Fig. 10. Links to the two distant light sources to the back of the viewer are created first, and links to the other two light sources are created at lower levels of the hierarchy.

⁴Some rendering artifacts are visible on these images: they result partly from z-buffer accuracy problems and limited screen resolution, two issues completely independent from the algorithms described here. They can conceivably be eliminated through proper antialiasing or software z-buffer computation, but the images reflect what is obtained on screen in an interactive session.

All light sources have the same power, therefore the bound on the energy transfer is primarily driven by the distance to each light source. For this scene the hierarchy of clusters was automatically constructed using a modified hierarchy of bounding volumes currently under development.

VI. CONCLUSIONS AND FUTURE WORK

A unified algorithm has been proposed for the simulation of energy exchanges between diffuse surfaces, isotropic participating media and object clusters. This comprehensive algorithm can be applied in all fields that require accurate simulation of combined surface/volume illumination. It also provides an avenue for the development of efficient clustering algorithms, by establishing a correspondence between volumes and groups of surfaces.

Results were presented for two automatic clustering methods, showing that clustering can significantly accelerate radiosity calculations by effectively removing the need for a complete initial linking phase. The proposed algorithm completely integrates volumes and surfaces, and allows the quick addition of new cluster types thanks to its object-oriented nature.

Future directions include the study of better criteria to make linking decisions between clusters. In particular the derivation of tighter error bounds could dramatically reduce the number of links. More accurate methods are needed to distribute energy from a cluster to its elements (and reciprocally), taking into account for instance intra-cluster visibility. The development of efficient algorithms for the calculation of visibility and transmittance factors is another promising research direction. In particular we hope to use the automatic correspondence between surfaces and volumes to compute error bounds on transmittance estimates. In general the use of mixed representations combining surfaces and volumes will allow each representation to be used in specific areas of the computation. Finally, a number of cluster properties are in fact functions of direction in space, and could be stored as such. The use of such explicitly directional clusters allows constant-time linking operations, but requires important storage capabilities [32].

ACKNOWLEDGEMENTS

This research was greatly accelerated by the courtesy of several colleagues who kindly shared some of their code with the author. Much of the comprehension of hierarchical algorithms was gained by studying a program provided by Pat Hanrahan. The hierarchical subdivision code using a k - d tree is based on software developed by Kevin Novins, Jim Arvo and David Salesin. Many thanks to George Drettakis for numerous stimulating discussions and useful comments on this paper. The scene in Fig. 10 was assembled using pieces of the Berkeley Soda Hall model. Thanks to Seth Teller and all members of the UC Berkeley walkthrough group for sharing this model. The anonymous reviewers provided many helpful comments and helped to improve the presentation of the results.

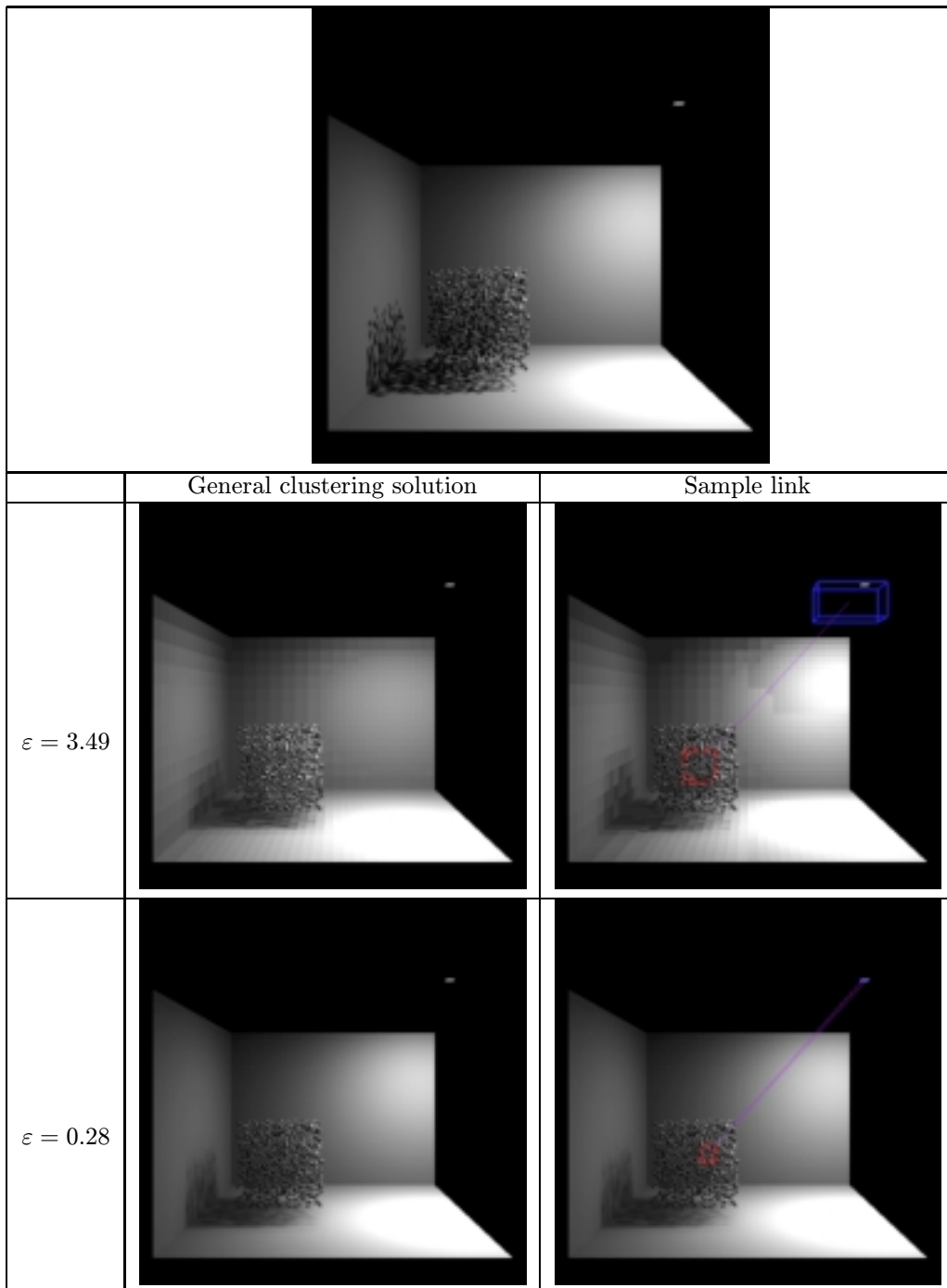


Fig. 9. General clustering solution for the same test scene as before. (top) reference solution (same as in Fig. 7). (left) general clustering solutions for two different error thresholds: 8.5s and 592s. (right) Sample links shown to illustrate the varying level in the hierarchy at which a link can be established, based on the subdivision error threshold.

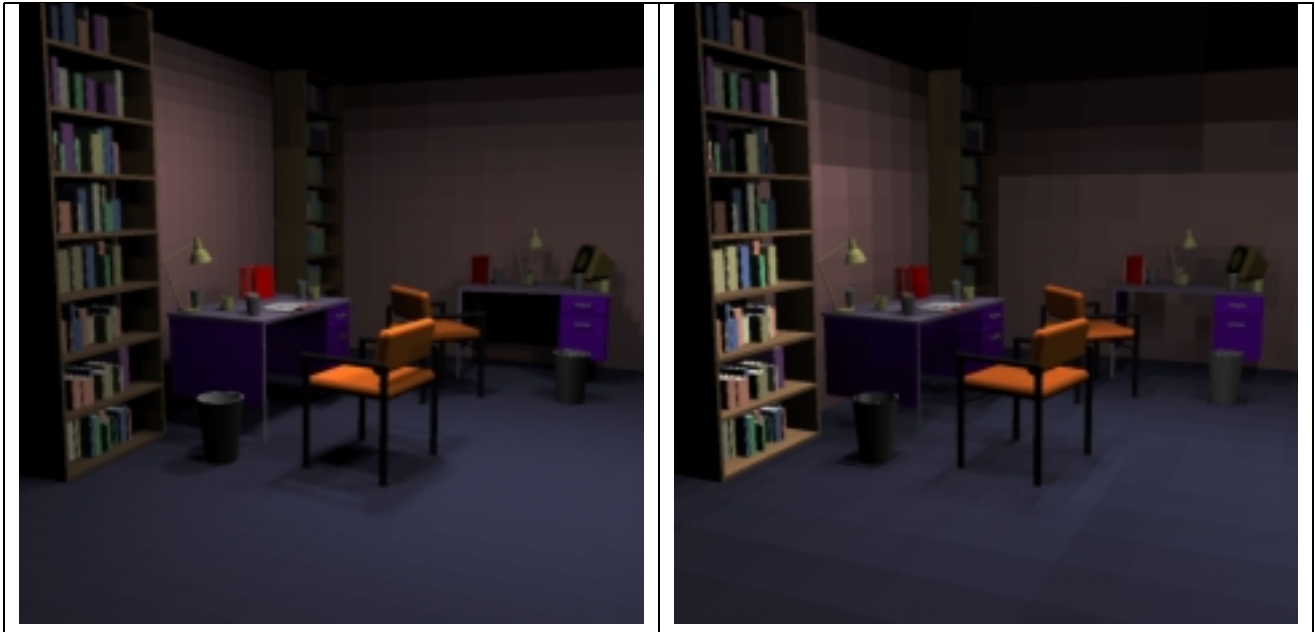


Fig. 10. Clustering solution for a complex environment (10,000 input polygons). (left) high-quality solution (directly rendered from the radiosity computation): 2 hours and 4 minutes. (right) approximate solution, obtained by using a higher error threshold and approximate visibility calculation using extinction coefficients: 28 seconds.

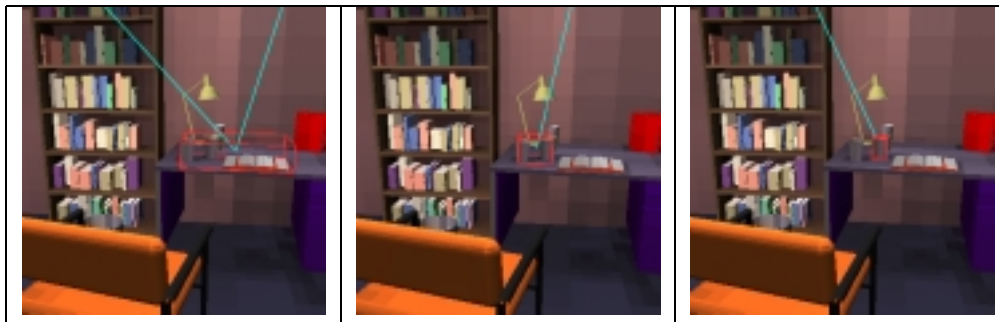


Fig. 11. For the solution shown above, links to the various light sources have been established with different clusters. From left to right we see that as the light source becomes closer, the link is created at a lower level of the cluster hierarchy. The three clusters shown are children of one another.

REFERENCES

- [1] Bruno Arnaldi, Xavier Pueyo, and Josep Vilaplana. On the division of environments by virtual walls for radiosity computation. In P. Brunet and F.W. Jansen, editors, *Photorealistic Rendering in Computer Graphics*, pages 198–205. Springer Verlag, 1993. Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, Spain, May 1991).
- [2] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '94* (Orlando, FL), pages 75–84. ACM SIGGRAPH, New York, July 1994.
- [3] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *Computer Graphics*, 25(4):51–60, August 1991. Proceedings SIGGRAPH '91 in Las Vegas (USA).
- [4] J.L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.
- [5] Neeta Bhate. Application of rapid hierarchical radiosity to participating media. In *Proceedings of AATRV-93: Advanced Techniques in Animation, Rendering, and Visualization*, pages 43–53. Bilkent University, July 1993.
- [6] Philippe Blasi, Bertrand Le Saec, and Christophe Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum*, 12(3):201–210, September 1993. Proceedings Eurographics '93.
- [7] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *Computer Graphics*, 22(4):65–74, August 1988. Proceedings SIGGRAPH '88 in Atlanta, USA.
- [8] Alain Fournier, Eugene Fiume, Marc Ouellette, and Chuan K. Chee. Fiat lux. Technical Report 90-1, University of Toronto, Dynamics Graphics Project, January 1990.
- [9] Reid Gershbein, Peter Schröder, and Pat Hanrahan. Textures and radiosity: Controlling emission and reflection with texture maps. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '94* (Orlando, FL), pages 51–58. ACM SIGGRAPH, New York, July 1994.
- [10] J. Goldsmith and J. Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications*, 7(5):14–20, May 1987.
- [11] Pat Hanrahan and David Saltzman. A rapid hierarchical radiosity algorithm for unoccluded environments. In C. Bouville and K. Bouatouch, editors, *Photorealism in Computer Graphics*. Springer Verlag, EurographicSeminars series, 1992.
- [12] 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).
- [13] Nicolas Holzschuch. Personal communication, 1994.
- [14] Hoyt C. Hottel and Adel F. Sarofim. *Radiative Transfer*. McGraw Hill, New York, 1967.
- [15] J. T. Kajiya and B. P. Von Herzen. Ray tracing volume densities. *Computer Graphics*, 18, July 1984. Proceedings SIGGRAPH '84 in Minneapolis (USA).
- [16] Arjan J. F. Kok. Grouping of patches in progressive radiosity. In *Proceedings of Fourth Eurographics Workshop on Rendering*, pages 221–231. Eurographics, June 1993. Technical Report EG 93 RW.
- [17] David Laur and Pat Hanrahan. Hierarchical splatting: A progressive refinement algorithm for volume rendering. *Computer Graphics*, 25(4):285–288, July 1991. Proceedings SIGGRAPH '91 in Las Vegas (USA).
- [18] 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), pages 67–74. ACM SIGGRAPH, New York, July 1994.
- [19] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '93* (Anaheim, CA, USA), pages 199–208. ACM SIGGRAPH, New York, August 1993.
- [20] Nelson Max. Efficient light propagation for multiple anisotropic volume scattering. In *Proceedings of Fifth Eurographics Workshop on Rendering*, Darmstadt, Germany, June 1994. Eurographics. Also appeared in SIGGRAPH'94 course notes No. 4.
- [21] Tomoyuki Nishita, Yasuhiro Miyawaki, and Eihachiro Nakamae. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *Computer Graphics*, 21(4):303–310, July 1987. Proceedings SIGGRAPH '87 in Anaheim (USA).
- [22] Chris Patmore. Illumination of dense foliage models. In *Proceedings of Fourth Eurographics Workshop on Rendering*, pages 63–71. Eurographics, June 1993. Technical Report EG 93 RW.
- [23] Holly Rushmeier. Solution methods for radiatively participating media, August 1992. SIGGRAPH Course notes (# 18, “Global Illumination”).
- [24] Holly Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proceedings Graphics Interface '93*. Morgan Kaufmann publishers, 1993.
- [25] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. *Computer Graphics*, 21(4):293–302, July 1987. Proceedings SIGGRAPH '87 in Anaheim (USA).
- [26] Georgios Sakas and Matthias Gerth. Sampling and anti-aliasing of discrete 3-d volume density textures. *Eurographics '91*, pages 87–102, September 1991. Proceedings Eurographics '91.
- [27] Hanan Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, Massachusetts, 1990.
- [28] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corporation, New York, third edition, 1992.
- [29] François Sillion. Clustering and volume scattering for hierarchical radiosity calculations. In *Proceedings of Fifth Eurographics Workshop on Rendering*, Darmstadt, Germany, June 1994. Eurographics.
- [30] 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).
- [31] 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.
- [32] François Sillion, George Drettakis, and Cyril Soler. A clustering algorithm for radiance calculation in general environments. In *Sixth Eurographics Workshop on Rendering*, Dublin, Ireland, June 1995.
- [33] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann publishers, San Francisco, 1994.
- [34] 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), pages 435–442. ACM SIGGRAPH, New York, July 1994.
- [35] Brian E. Smits, James R. Arvo, and David H. Salesin. An importance-driven radiosity algorithm. *Computer Graphics*, 26(4):273–282, July 1992. Proceedings of SIGGRAPH '92 in Chicago (USA).
- [36] Lee Westover. Footprint evaluation for volume rendering. *Computer Graphics*, 24(4):367–376, August 1990. Proceedings SIGGRAPH '90 in Dallas (USA).
- [37] H. Xu, Q. Peng, and Y Liang. Accelerated radiosity method for complex environments. *Eurographics '89*, pages 51–61, September 1989.

APPENDIX

I. PSEUDO-CODE FOR GENERIC OPERATIONS

A. Gathering operations

The gathering operation is performed during the solution stage, after the set of links has been refined. In our implementation, this operation is performed by calling a generic **Gather** function recursively operating on an H-element. The function computes the irradiance received at the current level across all links, in the following manner:

```

Gather ( H-element  $p$  )
  for each link  $l$  arriving at  $p$  do
    GatherIrradiance(  $p, l$  )
  end for
  for each child  $c$  of  $p$  do
    Gather(  $c$  )
  end for

GatherIrradiance ( H-element  $p$ , Link  $l$  )
   $p$ .Irradiance  $\leftarrow p$ .Irradiance +  $l$ .Source.Radiosity
   $\times l$ .FormFactor

```

The functions above suffice for surface radiosity or isotropic clustering. However in the general clustering case the irradiance received across each link comes from a different direction, and cannot simply be added to a global irradiance value. Therefore it must be distributed to each cluster's contents. This is accomplished by overriding the **GatherIrradiance** procedure for clusters only:

```

GatherIrradiance ( H-Cluster  $p$ , Link  $l$  )
  Spectrum  $Irr \leftarrow l$ .Source.Radiosity  $\times l$ .FormFactor
  /  $p$ .Scalefactor
  Point  $Q = l$ .Source.SamplePoint
  DistributeIrradiance (  $p, Irr, Q$  )

DistributeIrradiance ( H-Cluster  $p$ , Spectrum  $Irr$ , Point  $Q$  )
  for each polygon  $c$  in  $p$  do
     $c$ .Irradiance  $\leftarrow c$ .Irradiance +  $Irr \times \mathbf{ReceiverFactor}(c, Q)$ 
  end for
  if  $p$  is not a leaf
    for each Cluster child  $c$  of  $p$  do
      DistributeIrradiance (  $c, Irr, Q$  )
    end for
  end if

```

Note that all radiosity values are defined with the **Spectrum** type. This simply indicates that they depend on wavelength. In our implementation the Spectrum type is represented by an array of floating point values. The **ReceiverFactor** term in this code is the orientation factor (cosine of the incident angle) described in Section II-C.2.

B. Push - Pull

Here we describe the Push-Pull procedure that redistributes contributions received at all levels of the hierarchy to ensure that each H-element has an accurate view of its radiosity, including what was received by its ancestors and children. As explained in the text, irradiance is pushed down the hierarchy, until a leaf H-element is encountered. In the case of a leaf surface, the local reflection operator is applied to transform irradiance into radiosity:

```

Spectrum PushPullRad ( H-element p, Spectrum IrrDown )
  Spectrum RadUp
  if p is a leaf
    RadUp ← GetLeafRadiosity( p , IrrDown )
  else
    for each child c of p do
      RadUp ← Radup + c.AreaFactor
        × PushPullRad( c , p.Irradiance + IrrDown )
    end for
    RadUp ← RadUp / p.AreaFactor
  end if
  p.Radiosity ← RadUp
  return RadUp

Spectrum GetLeafRadiosity ( H-element p, Spectrum IrrDown )
  return (WhiteSpectrum - p.Reflectance) × p.Exitance +
    p.Reflectance × ( p.Irradiance + IrrDown )

```

However for leaves of the cluster hierarchy, irradiance must be pushed down to the surfaces contained in the cluster, if any. In addition, the generic local reflection operator is also called to account for the possible volume scattering occurring in the cluster:

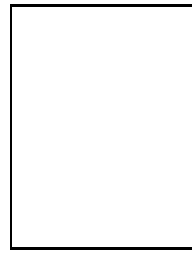
```

Spectrum GetLeafRadiosity ( H-Cluster p, Spectrum IrrDown )
  Spectrum RadUp
  if p is empty
    return NullSpectrum
  else
    float localArea ← p.AreaFactor
    for each polygon c in p do
      localArea ← localArea - c.AreaFactor
      RadUp ← RadUp + c.AreaFactor
        × PushPullRad( c , p.Irradiance + IrrDown )
    end for

    Spectrum LocalRad ←
      GetLeafRadiosity( (H-element)p, IrrDown )
    RadUp ← RadUp + localArea*LocalRad
    RadUp ← RadUp / p.AreaFactor
    return RadUp
  end if

```

In this pseudo-code the area factor due to the local scattering volume is computed by subtracting the contribution of all polygons to the total area.



François Sillion studied at the Ecole Normale Supérieure in Paris, France, where he obtained a graduate degree in Physics and a PhD in Computer Science. After holding a post-doctoral position in the program of computer graphics of Cornell University, Sillion joined the french "Centre National de la Recherche Scientifique" (CNRS). He is since a member of the iMAGIS project, created at ENS in Paris and relocated in Grenoble in 1993. He also holds a part-time teaching position at Ecole Polytechnique in Palaiseau, France. His research focuses on the accurate simulation of complex phenomena and environments. He has recently published, with Claude Puech, a comprehensive book on radiosity: "Radiosity and Global Illumination". He is a member of ACM SIGGRAPH, EUROGRAPHICS, and the IEEE Computer Society, and current chair of the Eurographics working group on rendering.