The Visual Computer

A case study towards validation of global illumination algorithms: progressive hierarchical radiosity with clustering

Karol Myszkowski[1],
Tosiyasu L. Kunii[2]

[1] Computer Graphics Laboratory, Aizu University,
Aizu Wakamatsu, 965-8580 Japan
e-mail: k-myszk@u-aizu.ac.jp
[2] Computing Science Research Center, Hosei
University, Koganei City, Tokyo, 184-8584 Japan
e-mail: kunii@k.hosei.ac.jp

In this paper, we present an efficient global illumination technique, and then we discuss the results of its extensive experimental validation. The technique is a hybrid of cluster-based hierarchical and progressive radiosity techniques, which does not require storing links between interacting surfaces and clusters. We tested our technique by applying a multistage validation procedure, which we designed specifically for global illumination solutions. First, we experimentally validate the algorithm against analytically derived and measured real-world data to check how calculation speed is traded for lighting simulation accuracy for various clustering and meshing scenarios. Then we test the algorithm performance and rendering quality by directly comparing the virtual and real-world images of a complex environment.

Key words: Clustering – Hierarchical radiosity – Progressive refinement – Experimental validation – Global illumination

Correspondence to: K. Myszkowski

The Visual Computer (2000) 16:271–288

1 Introduction

It is relatively easy to use commodity rendering techniques to create images that look great; however, it is much more difficult to create images that match the appearance of an actual space. Good matching of synthetic images and their real-world counterparts is desirable in many engineering applications, including architecture, lighting, and interior design. The basic precondition to achieve a high level of realism, which may be required in such applications, is physically based lighting simulation. Global illumination techniques are specifically designed for this purpose. However, to make computation tractable in practical applications, many simplifying assumptions are usually introduced to the underlying physical models. Because analytic evaluation of such simplifications and interactions between them is generally impractical, the correctness of a given technique must be checked experimentally by comparison of simulation results to some reference data. For example, distribution of illumination at some predefined points derived analytically or measured experimentally (this is usually performed in specialized measurement rooms) can be used to validate the lighting simulation part of a rendering algorithm. An effective way to test complete rendering algorithms [including the image display procedure on the CRT device (Tumblin and Rushmeier 1993)] is a direct comparison of virtual and real-world images. Unfortunately, such experimental validation has almost never been performed for existing global illumination solutions. In this research, our goal is to develop an efficient global illumination technique whose lighting simulation correctness is validated experimentally, and which is capable of producing images of quality comparable to that of photographs. Radiosity techniques show their high potential in photorealistic image synthesis and solving the global illumination problem for environments with dominating Lambertian reflectance properties. In particular, recently developed hierarchical radiosity algorithms with clustering are very efficient in terms of processing speed, which potentially could make them suitable for rendering scenes of quite involved complexity. However, in practical applications, the achieved scene complexity is severely limited by the large memory required to store the huge number of links, which are used to guide energy exchange between surfaces (clusters) at various levels of hierarchy. In traditional hierarchical radiosity methods, all those links should be stored simultaneously in the memory. Furthermore, the *initial linking* phase, which must be completed before lighting energy can be transfered within the scene, affects the progressivity of computations because meaningful partial solutions are provided to the user only with a substantial delay.

In this paper, we present a hybrid of progressive and hierarchical radiosity with clustering, which avoids many of the drawbacks of traditional algorithms. The proposed technique satisfies the following requirements of high practical importance:

- Low growth rate of calculation time for increasing complexity of scenes
- Low memory requirements (in particular, links need not be stored)
- Progressive refinement, i.e., quickly producing approximate, but meaningful results, which converge to an accurate solution
- Low sensitivity to the style of scene modeling (the user is not forced to adjust his modeling habits to the specific requirements of the algorithm).

We validate our algorithm experimentally by comparison of simulation results to analytically derived data (which are available for simple scenes only), as well as to real-world measurements for more involved environments. We also compare virtual and real-world images for a case-study environment of high complexity. To our knowledge, such a wide spectrum of validation experiments had never been performed in the context of hierarchical and progressive radiosity techniques. We believe that our validation procedure can be useful to test other global illumination techniques as well.

In the following section, we discuss existing hierarchical and cluster-based radiosity solutions. The new algorithm is then described in Sect. 3. We provide the results of our validation experiments in Sect. 4, and we show rendering performance of our technique in Sect. 5. Finally, we conclude this work.

2 Previous work

The radiosity method and its basic variants, such as progressive radiosity and hierarchical radiosity techniques, are described in detail in several textbooks (Cohen and Wallace 1993; Sillion and Puech 1994). We now review some basic definitions relevant to this research, and discuss more recent developments. The numerical methods applied to solve a linear system of equations resulting from the radiosity formulation are based on two basic types of iterative solutions: the *gathering iteration* and the *shooting iteration*. In the gathering iteration, light incoming from all other surfaces is summed up to compute the radiosity of a given patch. In the shooting iteration, the lighting energy of a given patch is distributed toward all other patches, and their radiosity is updated based on the light arriving from the shooting patch.

The *basic (nonhierarchical) radiosity* formulation relies on energy exchange between predefined patches via gathering or shooting iterations. Although patches might be subdivided into finer elements [by the so-called *substructuring* technique (Cohen et al. 1986)] the actual solution is performed only at the coarsest level of patches. Those elements do not take any active part in the radiosity solution, and they are used mostly to improve the quality of shading during rendering. The complexity of nonhierarchical solutions is $O(n^2)$ (Hanrahan et al. 1991), which means that the calculation load increases quadratically with the number of patches n.

The *hierarchical radiosity* (HR) formulation involves the computation of energy exchanges between various levels of the patch hierarchy. For a given pair of interacting surfaces, the choice of the level of hierarchy (effectively, the size of patches) can be based on the value of the form factors, the amount of exchanged energy, the perceivable impact of such an energy exchange on the quality of the rendered image, and so on. The complexity of the hierarchical solution is $O(m^2+n)$ (Smits et al. 1994), where m is the number of input surfaces.

In general, HR techniques strongly outperform their nonhierarchical counterparts because usually $m \ll n$. However, the well-known drawback of practical HR algorithms is the strong dependence of their performance on geometric model preparation, which can strongly affect m (the same model can be built of vastly different numbers of polygons, depending on the modeling software or user style). In addition, disproportions between neighboring patches that are imposed by the hierarchy usually result in shading artifacts when the reconstructed radiosity is rendered (Smits et al. 1992; Lischinski et al. 1993; Smits et al. 1994). Solutions proposed to improve the quality of the shading usually significantly reduce the overall HR performance. For example, Smits et al. (1992) force finer subdivision than required by the assumed accuracy of the energy transfer for the visible part of the scene (their algorithm is view dependent). Lischinski et al. (1993) propose two passes: (1) a *global pass*, which uses HR to compute radiosity solution, and (2) a *local pass* to refine the radiance distribution locally on each surface. A similar approach was applied by Smits et al. (1994), but they found it very time consuming and deferred further research on the acceleration of the local pass.

The existing HR algorithms can be classified as the gathering HR (GHR) and the shooting HR (SHR), depending on the iteration type that is used to simulate light interaction between surfaces. The vast majority of research projects conducted so far focused on GHR techniques (Hanrahan et al. 1991; Lischinski et al. 1993; Holzschuch et al. 1994; Smits et al. 1994; Teller et al. 1994; Sillion 1995; Gibson and Hubbold 1996; Willmott and Heckbert 1997). To the knowledge of the authors, only two works on SHR algorithms (Myszkowski and Kunii 1995; Stamminger et al. 1998) have been presented. In this research, we pursue the SHR approach; we are motivated by numerous drawbacks of the GHR techniques, which we now discuss.

The very principle governing the GHR techniques is the strict control of energy transfers effected by introducing links between interacting surfaces and refining these links until the energy sent through each link falls below a predefined threshold. The storage of links and related data structures requires a huge memory, e.g., 44 MB and 1.1 GB for medium complexity scenes composed of 1000 polygons (Willmott and Heckbert 1997) and 7054 polygons (Teller et al. 1994), respectively. The problem of link storage is even more dramatically exposed for higher-order hierarchical methods (Cohen and Wallace 1993). The required memory for such methods is beyond the limits acceptable in practical applications involving complex scenes [e.g., 550 MB were required for a scene composed of 1000 polygons (Willmott and Heckbert 1997)], and because of that, we do not discuss these methods further in this paper.

Another link-related drawback of GHR algorithms is the cost of initial linking, which is performed before any energy transfer is calculated. The initial linking delays the first possible preview of the approximated image, which affects the progressivity of GHR algorithms. Gibson and Hubbold (1996) report a case for which the standard GHR algorithm required 37 min for the initial linking, but only 92 s for the link refinement and solution convergence. Similar results have been independently reported by Smits et al. (1992) and Lischinski et al. (1993). Holzschuch et al. (1994) propose a lazy initial linking technique, which accelerates this phase by ignoring all links transferring low energy, but then the intermediate solutions may be visually affected because lighting energy may not be distributed to some scene regions.

In a traditional HR algorithm, originally input polygons undergo substructuring, but always remain atop the hierarchy (which makes HR performance very dependent on the geometric model representation). Clustering (grouping) of input polygons originally proposed for non-HR techniques (Kok 1993; Rushmeier et al. 1993) removed such a limitation. Within the GHR framework, clustering enables the building of an efficient and complete hierarchy (with the root cluster containing the whole scene) by recursive grouping of polygons and clusters (Smits et al. 1994; Sillion 1995; Gibson and Hubbold 1996). Obviously, clustering significantly reduces the problems of initial linking and link storage, which in turn makes it possible to process significantly more complex scenes (Gibson and Hubbold 1996). However, the problems with links do not vanish completely because each gathering iteration of the GHR solution still involves all links. Thus, clustering pushes the link problems to a quantitatively different level, but GHR with clustering remains a very memory-intensive technique.

2.1 Discussion

Let us examine more closely how links are actually used and refined in the GHR framework. Let us first consider the degenerate case of point light sources, as are often used in scene models. Clearly, there is no advantage of link storage for such lights, because after the link refinement in the first iteration, further refinement becomes impossible as there is no way that indirect light can contribute to these links. Not only that, in the case of area primary light sources, the contribution of indirect light is usually rather small compared to its direct counterpart, so those links are very rarely refined. In practice, a similar observation can be made for a significant percentage of links attached to secondary emitters (passive surfaces). For example, Holzschuch et al. (1994) report that between 65% and 95% of the links have not been refined for a number of scenes that they tested. This means that links serve mostly as passive pipes across all gathering iterations of the GHR algorithm.

Now, let us consider replacement of the gathering iteration by the shooting iteration within the HR framework. Clearly, for point light sources and area light sources with small reflectance values, links to other surfaces could be created once and then discarded completely. For area light sources with significant reflectance properties, the links could be created at the beginning of the simulation to distribute direct light, and afterwards they could be temporarily discarded. If significant indirect lighting energy were to accumulate for such light sources, then the links could be recomputed just to distribute unshot energy and then be discarded again. A similar approach could be taken to process all surfaces, i.e., to compute the links only if significant unshot energy must be distributed. Obviously, in all cases, the links are created at an appropriate level of hierarchy as required by HR algorithms.

An approach combining the strengths of the shooting iteration and the HR framework (with a limited hierarchy) is proposed by Myszkowski and Kunii (1995). In our technique, link storage was not required, and the computation progressivity was improved because the initial linking phase could be avoided, and the most important energy shooters could be selected in every iteration. Stamminger et al. (1998), who consider shooting iteration within the full-fledged HR framework, have shown that, while some links must be recomputed, the number of such links decreases exponentially if only unshot energy is considered. They also show theoretically that their SHR algorithm with clustering converges to the correct solution.

In this work, we build upon our earlier technique (Myszkowski and Kunii 1995) and focus on the following issues of high practical importance.

- *Improvement of the spatial locality of clusters.* Current clustering techniques are limited to the original input surfaces of the model, and they ignore their locality in the space. For example, in architectural applications, many sliverlike patches are often modeled; they have significant lengths with respect to the scene size, but their contributions to the energy exchange is rather small. This makes them perfect candidates for clustering. The question that arises is how to shape the resulting clusters well to improve the accuracy of energy transport. We address this issue in Sect. 3.1.
- *Introduction of a self-correcting mechanism of the energy transfer accuracy.* Such a mechanism is inherent for the GHR approach, but is missing in the SHR algorithm (Stamminger et al. 1998). In Sect. 3.3 we propose such a gathering-style correcting mechanism to improve the accuracy of the shooting iteration within the hierarchical framework.
- *Experimental validation of the SHR algorithm with clustering.* In Sect. 4, we describe our validation procedure, which makes it possible to evaluate our technique performance and accuracy. Such an evaluation complements the theoretical analysis of the convergence rate (Stamminger et al. 1998), which is based on very conservative assumptions (worst-case studies).

3 Overview

In the following description of our global illumination solution, we focus on indirect lighting computation. Lighting simulation starts from computing the direct lighting at the vertices of the initial mesh. Adaptive mesh subdivision is performed to improve the quality of shading and accuracy of energy transfer. Indirect lighting computations are performed next.

Secondary lighting is first calculated for initial mesh patches (subdivision performed due to primary lights is ignored at this stage), and then, if necessary, the solution is refined by adaptive subdivision of mesh patches exhibiting high gradients of luminance due to indirect light. Usually, the quality of shading is quite good after the first phase, so, in practice, the second phase can very often be omitted, unless subtle shading effects must be produced, e.g., shadows resulting from secondary lights in the regions that are not directly illuminated.

3.1 Clustering

As we discuss in Sect. 2, clustering is an important step toward making HR techniques practical for handling complex scenes. A traditional approach to clustering relies on grouping input geometrical elements (usually polygons) and uses some clustering criteria such as proximity of locations in space, similar orientation, and so on (Sillion and Drettakis 1995; Christensen et al. 1997).

The resulting clusters are then very sensitive to the way in which the original input geometry is represented. To alleviate this problem, we first subdivide the surfaces during initial meshing, and then

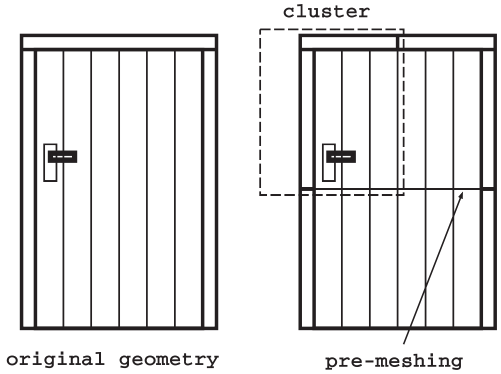


Fig. 1. Mechanics of clustering

group them back during clustering. The problem of long, thin polygons (common in architecture applications) should be recalled here. Instead of grouping input polygons, a better locality of clusters can be expected when the slivers are subdivided into better-shaped polygons and then grouped into multiple local clusters (Fig. 1). In such a case, the clustering does not rely on the input geometry, but is performed on the basis of local position in the scene. The locality of the resulting clusters increases the number of possible high-level energy transfers between such clusters compared to the clusters based on input geometry. The better aspect ratio of mesh patches improves the accuracy of form factor calculation, as well as the quality of shading. The mesh constitutes a good basis for displaying images reporting the current status of computation immediately upon user request. Of course, the cost of subdividing the polygons must be amortized by the gains of clustering.

It is worth noting that the proposed approach to clustering combines, in a unified framework, the clustering of input polygons and the hierarchy of the subdivisions of the input polygons. In traditional techniques, clustering and hierarchical subdivision constitute two independent processes, depending heavily on the input geometry. In our case, the hierarchy is not built upon the input geometry, but rather uses the coherence of positioning in the space. This means that a single cluster can be built from:

- Original input polygons – as in traditional clustering.
- Selected mesh patches belonging to the same input polygon – as in the hierarchical radiosity algorithm.
- Selected mesh patches belonging to different input polygons, i.e., portions of the input polygons tessellated by mesh – this functionality is unique to our approach.

Many authors report problems with the shading artifacts that appear because of poor grading of the surface size imposed by the hierarchy (Smits et al. 1992; Lischinski et al. 1993; Smits et al. 1994). We decided to flatten our hierarchy to avoid the shading artifacts, especially since our goal is to make it possible to render good-quality illumination maps at any stage of the computation; these maps can be selected interactively by the user. The initial meshing is the first step in this direction. Tessellation of big surfaces into a uniform mesh makes the energy transferred by mesh patches smaller and better balanced. Furthermore, we limit the hierarchy to two levels (clusters and mesh patches) in the course of the radiosity solution for indirect lights. When further subdivision is required because of error bounds imposed on the energy transfer between patches, we adaptively increase the accuracy of the form factor calculations instead of extending the number of hierarchy levels.

Of course, the flattened hierarchy may decrease the algorithm performance for some scenes compared to that of a fully hierarchical approach. Nonetheless, our approach with the flattened hierarchy of light interactions is also supported by the experimental results published in the literature, e.g., Lischinski et al. (1993) report an average depth of hierarchy in the range 1.31–2.04 for scenes they tested; similar results are obtained by Holzbuch et al. (1994).

In both cases, the hierarchy was built for both primary and secondary light sources. In our approach, an even lower depth of hierarchy can be expected because the input surfaces are premeshed and hierarchical interactions between surfaces involve only indirect lighting.

In our approach, the clustering is performed automatically. A grid of voxels is used to group the mesh patches into clusters. All patches belonging to a cluster are located in the same voxel and have similar orientations in the space within the specified tolerance of the normal vector direction. The tolerance of variation of normal vectors trades the number of clusters for quality of shading (illumination of the cluster is pushed down to the patches and their orientation is ignored). In order to avoid assignment to multiple clusters of a single patch spanning the border between voxels, the location of the center of gravity is considered. The number of voxels that determines the size of the clusters can be defined by the user explicitly, or implicitly as the average number of mesh patches in the cluster.

3.2 Shooting iteration

In our technique, energy is transferred exclusively via the shooting iterations. This requires the data structures that combine records typical of progressive radiosity (PR) (Cohen et al. 1988) (the total and unshot energies are stored), and for hierarchical techniques (the record of light interaction in the scene is maintained on the level of clusters and mesh patches). For every patch i, records with the total L_i and unshot ΔL_i luminances are stored. The unshot luminance of a cluster ΔL_C is derived as the area-weighted average of ΔL_i for all patches k assigned to the cluster:

$$\Delta L_C = \sum_{i=1}^{k} \Delta L_i \frac{A_i}{A_C}, \qquad (1)$$

where A_i and A_C are the surface areas of the ith patch and of all patches in the cluster, respectively. This approach corresponds to energy pulling up in the GHR algorithm, but involves the unshot energy. The lighting energy exchanged between clusters is stored as illumination I_C, which is pulled down to each cluster's patches by the following sequence of assignment operations:

$$L_i := L_i + I_C \frac{\rho_i}{\pi}, \qquad \Delta L_i := \Delta L_i + I_C \frac{\rho_i}{\pi},$$
$$I_C := 0, \qquad (2)$$

where ρ_i is the diffuse reflection coefficient of the ith patch.

For the nonhierarchical PR algorithm, the total and unshot energies are implicitly updated during lighting calculations, but the hierarchical lighting interaction requires an explicit energy update between the levels of hierarchy (the push-pull procedure). In our algorithm, the cluster illumination I_C is pulled down to the patches (2), and then the updated unshot energy of patches is pushed up to ΔL_C (1). The cost of traversing our flat hierarchy is relatively small, but performing this procedure for all clusters after every shooter is considered would be impractical. However, the unshot energy cannot be updated too lazily because the progressivity of the algorithm may be affected (the choice of the most important shooter). In our algorithm, the push-pull procedure is performed each time the user-predefined percentage of the unshot energy for the whole scene is processed. The push-pull procedure is also executed when the image display is requested by the user, or for a cluster that is selected as the shooting cluster (and its ΔL_C must correspond to its analog for the patches ΔL_i).

Secondary lighting simulation is initialized for every cluster by computation of ΔL_C in (1) based on direct illumination of all patches i belonging to a given cluster. The iterative solution proceeds until the unshot energy falls below a threshold predefined by the user. At every step, the choice of the most important shooter (a cluster or a mesh patch that does not belong to any cluster) is based on its unshot energy $\Delta L_C A_C$ (candle power). In the following section, we propose a novel hierarchy refinement scheme within the framework of the shooting iteration, which is performed to increase the accuracy of lighting transfer computations.

3.3 Hierarchy refinement with energy transfer correction

One of the serious problems with the SHR approach is the lack of a self-correcting mechanism of the energy transfer accuracy. In every iteration, only *unshot* energy is considered, and if this unshot energy is always below a certain threshold for all iterations, the refinement of the hierarchy level for a given pair of patches exchanging energy might never be triggered, although the total energy exchanged between the patches can be well over the threshold. Similarly, if the hierarchy level is refined at a certain point in the computations, the refinement affects only energy transferred in the current and subsequent iterations, while energy shot in the preceding iterations has been distributed at coarser levels of hierarchy (which means with lower accuracy). Obviously, the resulting errors in energy distribution propagate across all surfaces in the scene. In the GHR framework, this problem is automatically avoided because, in every iteration, the whole radiosity accumulated by patches is distributed through the link structure connecting all surfaces in the scene. In this section, we propose a gathering-style mechanism that improves the accuracy of energy transfers for the SHR approach.

The flattened hierarchy of light interactions imposes two basic energy transfer modes: cluster→cluster, and patch→patch. The cluster→cluster mode is allowed when the resulting illumination of a receiver is smaller than a predefined threshold value ΔI_{max}. The lower bound on the required distance between the clusters is evaluated as:

$$R = \sqrt{\frac{L_C A_C}{\Delta I_{max}}},$$

where the total cluster luminance L_C is derived as:

$$L_C = \sum_{i=1}^{k} L_i \frac{A_i}{A_C}.$$

The distance R is expressed as r_v in terms of voxels, which are used to define clusters. If (i, j, k) denote the indices of the voxel where the shooter is located, then the cluster→cluster energy is transferred for the voxels (x, y, z) that satisfy the condition

$$\max(|x-i|, |y-j|, |z-k|) \geq r_v. \tag{3}$$

Otherwise, the energy transfer mode patch→patch is applied. The distance r_v is stored and compared in subsequent iterations with its updated value calculated for current values of $L_C A_C$. When previous and current values of r_v are equal, the unshot energy $\Delta L_C A_C$ is transferred in the same way as for the last shooting. Otherwise, for voxels that do not satisfy the condition (3) for the current r_v, but satisfy the condition for the previous r_v, the level of hierarchy of interacting surfaces is refined, and the energy transfer cluster→cluster is replaced by the patch→patch scheme. Here, the energy transfer correction mechanism is introduced. Instead of shooting just unshot energy $(\Delta L_C A_C)$ between patches, the full energy transfer $(L_C A_C)$ is considered. This requires compensation for energy shot in the previous iterations by the source cluster to the receiver cluster. The following steps are performed for such energy correction.

1. Illumination of the receiving cluster I_C^r, which was the result of the cluster→cluster interaction in the previous iterations, is reconstructed. Our algorithm is deterministic, so we can repeat exactly the previous form factor calculations for the considered pair of clusters. The difference $(L_C - \Delta L_C) A_C$ is equal to the energy shot by the source cluster in the previous iterations.
2. The total L_i^r and unshot ΔL_i^r luminances stored for every patch of the receiver cluster are decreased by the value $I_C^r \frac{\rho_i^r}{\pi}$.
3. Lighting interaction between patches is performed; the total energy $L_i A_i$ of the source patch i is shot to replace the unshot energy $\Delta L_i A_i$, which is used when the level hierarchy does not require refinement.
4. The push-pull procedure is executed to update ΔL_C^r and L_C^r for the receiver cluster.

The scheme of refinement is equivalent to the link refinement in GHR algorithms, but steps 1, 2, and 4 are not required in the GHR framework, in which all links are stored. The cost of the extra steps is rather small because the results of the visibility calculations (the most time-consuming part of the form factor computation) at step 1 are reused when the energy is transferred between patches in step 3.

The proposed algorithm offers a limited refinement of the hierarchy within the SHR framework. Such a refinement scheme, which always involves the total energy exchanged between a pair of clusters (as in GHR algorithms), improves the overall accuracy of the energy transfer by reducing the accumulated error inherent in the shooting iteration. This limited self-correction mechanism is unique to our approach, and cannot be obtained with the shooting iteration proposed by Stamminger et al. (1998). While, in our current implementation, the self-correction mechanism is available only for a limited hierarchy, its extension to a full SHR approach is straightforward.

4 Experimental validation

We validated our hierarchical technique extensively with the shooting iteration, focusing both on the accuracy of lighting simulation and on the quality of rendering. In this section, we report results of our experiments. We have chosen very simple scenes as basic test data because the correct solution for such scenes can be found analytically (although, for an empty cube, the direct numerical integration of the rendering equation was necessary as well). We also compare the accuracy of our algorithm against the measurement data for more complex environments. However, in this case, we are not able to estimate

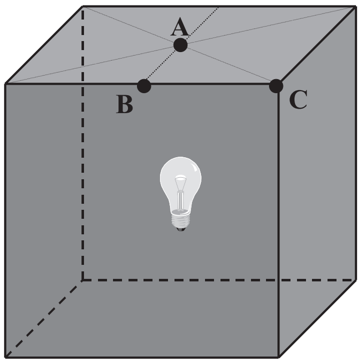


Fig. 2. Location of sample points used for comparison of simulation accuracy in Table 1

the measurement error precisely, a problem that does not exist for the analytically derived data. Finally, we compare the quality of an image generated by our technique and the corresponding photograph of a real-world architectural object. All timings reported in this paper were measured on a MIPS R10000 195 MHz processor.

4.1 Comparison with theoretical results

4.1.1 Empty cube test

The first scene used for comparison is an interior of an empty cube (10 m × 10 m × 10 m) with one point light source at its center (Fig. 2). The luminous intensity of the light source is equal to 50 000 cd, resulting in a direct illumination level of 2000 lx at the nearest point on the walls. The wall material is white with a diffuse reflectivity of 2/3, so that an indirect component constitutes a greater part in the full illumination. Table 1 shows the theoretical results derived for three points inside the cube, which are shown in Fig. 2.

Computation of direct lighting for these three points is straightforward, and we got a perfect match between the theoretically derived illumination and the simulated illumination. In further considerations, we present the simulation errors in the context of total and indirect lighting. However, in our discussion, we focus mostly on the errors for indirect lighting only. They are more relevant to our hierarchical radiosity with the shooting iteration. The selection of locations of sample points at which we measured the accuracy of our technique (refer to Fig. 2) proved to be quite difficult, and we believe that other practical radiosity implementations share many similar problems that we discuss later.

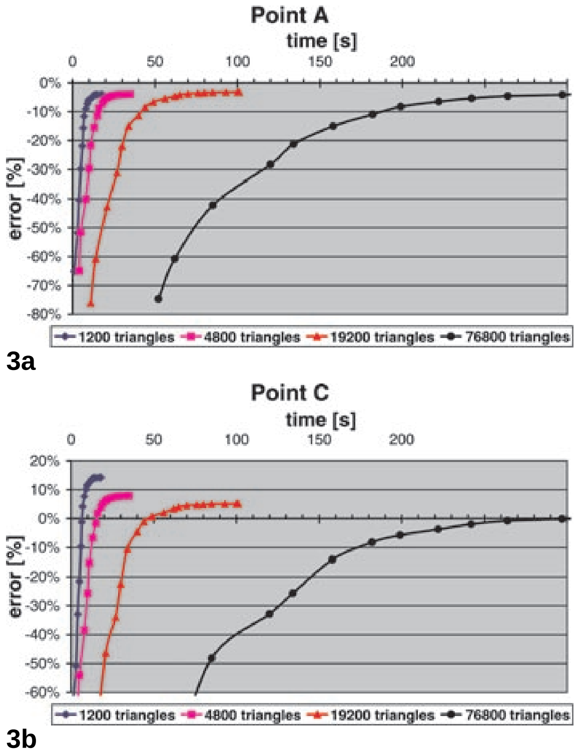
Table 1. Theoretically derived illumination values for selected points in the empty cube scene

Sample	Theoretically derived results	
point	Direct illumination (lx)	Indirect illumination (lx)
A	2000.0	2205.1
B	707.1	1954.5
C	384.9	1444.5

We prepared two sets of tests to measure the convergence of the radiosity solution as a function of time at the three sample points. The goal of the first test was to investigate how the clustering compensates for the increasing mesh complexity. We assumed that the cluster size is fixed to 1% of the wall surface area. Graphs in Fig. 3 summarize the obtained results. As can be seen, the speed of solution convergence scales quite well with the increasing number of triangles. However, the scaling is affected by the intracluster computation (a patch→patch energy transfer is always performed), which has quadratic complexity in respect to the number of triangles within a cluster. In the cube scene, clusters located along the cube edges may involve intensive computations for heavily populated clusters.

The goal of the second experiment was to check how the simulation error is affected by the variable cluster size. We fixed the number of triangles to 19 200 (3200 triangles per wall). Graphs in Fig. 4 summarize the obtained results. In general, the accuracy of the computations increases when the cluster size is reduced. However, the proper interpretation of results depicted in Fig. 4 can sometimes be quite involved because of the reasons discussed later. Furthermore, the convergence speed for big clusters, e.g., 4% of the wall surface area, can be worse than that for smaller clusters such as 1% of the wall surface area. This is caused by the costs of the intracluster energy transfer computation. Thus, an optimal setting of the cluster size exists, but might be substantially different for various scenes, and it is quite difficult to predict in advance.

Graphs in Figs. 3 and 4 are complemented by the results shown in Tables 2 and 3, which should be



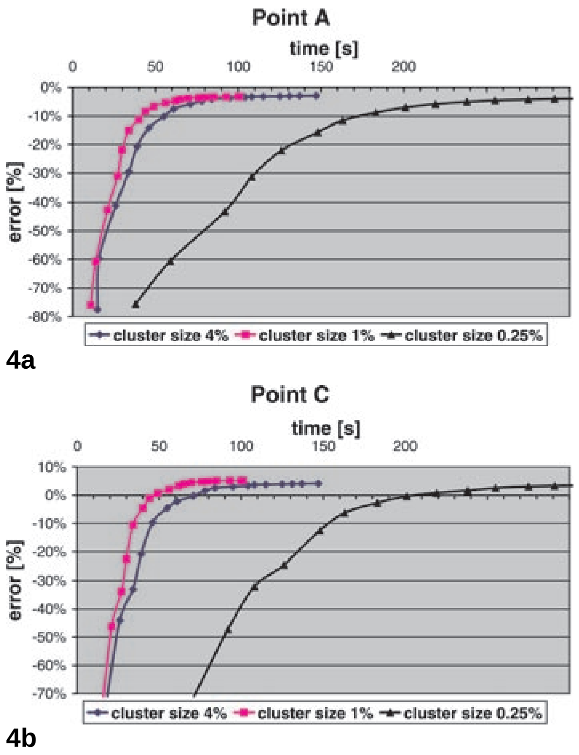


Fig. 3. Simulation error at points A and C for the cluster size fixed to 1% of the cube's wall surface area, and for various mesh tessellations. The relative error measured in respect to indirect lighting only is shown

Fig. 4. Simulation error at points A and C for mesh tessellation fixed to 19 200 triangles and for various cluster sizes. The relative error measured in respect to indirect lighting only is shown

helpful in understanding of the simulation error behavior in the context of various clustering scenarios. We discuss the error issues for all three sample points separately because the specific considerations are different for each point type.

Computing the illumination at sample point A seemed to be the easiest task. However, even in this simple case we faced some problems. When walls of the cube are subdivided into an even number of triangles along every edge of the cube, then point A is located at the vertex shared by neighboring triangles in the middle of a wall. This situation is depicted in Fig. 5a. In our radiosity solution, illumination is computed for the weight center of every triangle. Then illumination at point A is derived as an average of the illumination of all triangles that share point A as a common vertex. From purely geometrical considerations, it can easily be seen that the illumination at point A (at which illumination achieves its maximal value) will be always underestimated. Since the discrepancy of illumination between the triangle center and its vertex (an extreme point within a triangle) usually achieves its maximal value (for the cube test this is always true because of the smooth lighting distribution), the measure of this discrepancy is a good estimate of the discretization error that results from the discrete formulation of the radiosity equation (Cohen and Wallace 1993). The obvious way to decrease this error is to reduce the size of the triangles. However, in the context of hierarchical radiosity, the wall region surrounding point A has only a small chance of triggering hierarchy refinement (in our experimental settings, the whole energy was transferred to this region via cluster→cluster interaction). Thus, when the cluster size is fixed, the discretization error at point A remains mostly intact. As can be seen from the data provided in Table 2 and Fig. 3a, the accuracy of the illumination

Table 2. Simulation errors of total and indirect illumination for the empty cube scene for various sizes of mesh elements

Mesh complexity	Relative simulation error (percentage)						Computation
	Point A		Point B		Point C		time
(number of triangles)	Total	Indirect	Total	Indirect	Total	Indirect	(min:s)
1 200	−2.14	−4.09	0.17	0.23	11.24	14.23	00:18
4 800	−2.02	−3.95	−0.25	−0.34	6.21	7.86	00:35
19 200	−1.69	−3.22	−0.96	−1.31	4.13	5.23	01:40
76 800	−1.74	−3.32	−0.81	−1.11	0.69	0.87	07:20

Table 3. Simulation errors of total and indirect illumination for the empty cube scene for various sizes of clusters

Cluster size	Number of triangles	Relative simulation error (percentage)						Computation
		Point A		Point B		Point C		time
(percentage of the wall area)	per cluster	Total	Indirect	Total	Indirect	Total	Indirect	(h:min:s)
4.00	144	−1.55	−2.95	−1.00	−1.36	3.26	4.13	00:02:27
1.00	32	−1.69	−3.22	−0.96	−1.31	4.13	5.23	00:01:40
0.25	8	−1.93	−3.68	−1.41	−1.92	2.98	3.78	00:06:27
0.06	2	−1.90	−3.63	−1.22	−1.67	3.20	4.06	00:59:57
No clusters	–	−1.83	−3.50	−0.59	−0.80	3.70	4.68	10:14:16

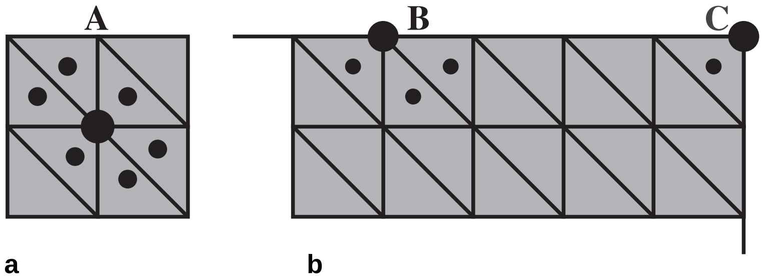


Fig. 5a,b. Location of sample points at which illumination values are computed during lighting simulation (marked by small dots). Illumination at mesh vertices is found by averaging results obtained for all triangles, which share a given vertex. This is also the case for sample points: **a** A; **b** B, and C, which are used to estimate the simulation accuracy in our tests

estimate at point A increases only slightly with the decreasing size of the triangles. This increase of accuracy can be attributed to better accuracy of the energy transfer along the cube edges because of intracluster computations, as well as the possibility of hierarchy refinement in these regions. For a fixed number of mesh triangles (in our tests we consider 19 200 triangles), the simulation error for indirect lighting stabilizes within the range of 3.5%–3.7% for small clusters (0.25% or 0.06% of the wall surface area) and for disabled clustering (refer to Table 3). Even smaller errors were obtained for bigger clusters (4% or 1% of the wall surface area). This can be attributed to the compensation of illumination underestimation (which is inherent in the location of point A in respect to the locations of radiosity sample points shown in Fig. 5a) by the overestimation of energy transfer resulting from the lower accuracy of cluster→cluster interactions.

Points B and C are extremely difficult cases for the radiosity computation because of the presence of singularities in the form factor numerical formulations (Cohen and Wallace 1993). The accuracy of the form factors rapidly deteriorates when surfaces are close to each other, and this is exactly the case for patches connected to the cube edges in our test data. The worst accuracy can be expected for point C located at a corner of the cube.

Points B and C are located at mesh vertices (similarly to point A), so their illumination is estimated by averaging the illumination at the weight centers of neighboring triangles, which are shown in Fig. 5b. For point B, a slight underestimation of the indirect illumination value can be observed. It is always below 2% (refer to Tables 2 and 3). In the remainder of the discussion, we focus on point C for which the maximum errors were obtained.

Illumination at point C is always overestimated, and the behavior of the simulation error in the context of clustering is quite different than that for the case of point A. Because of the proximity of the three walls, many important energy transfers are performed on the patch→patch basis, which means that the reduction of the patch size should reduce the discretization error and improve the simulation accuracy. Indeed, graphs in Fig. 3b and the accompanying data in Table 2 for the fixed size of cluster and variable resolution of mesh show quick reduction of the simulation error as the number of mesh elements increases. However, graphs in Fig. 4b and the accompanying data in Table 3 for fixed mesh show a significant overestimation (around 4%) of the illumination at point C, which cannot be reduced by decreasing the cluster size. Effectively, the accuracy of the energy transfer computation is increasing. However, this is not accompanied by a reduction of the discretization error.

The experiments show that clustering can make the interpretation of simulation results less obvious. In particular, the discretization error is affected not only by the mesh size, but also by the clustering parameters and hierarchy levels, which prevail when the light interaction of a given surface with the whole environment is computed. To reduce the influence of the discretization error on the overall simulation accuracy, we performed the energy gather step based on our radiosity solution for small differential regions surrounding points A, B, and C. This time, the best results were obtained for point A, and the worst for point C. In all cases tested, the simulation error was below 1% even for the coarse mesh built of 1200 triangles. The error obtained in such idealized conditions is very low. However, taking into account that a majority of applications refers to lighting reconstructed by mesh, the analysis of the total error (including the discretization error) that we performed in this section seems to be more relevant.

We used the cube test to check the influence of the cluster locality on the accuracy of lighting simulation. We investigated two approaches to clustering: (1) with premeshing of input geometry (as in Fig. 1), which resulted in grouping of well-shaped triangles, and (2) without premeshing, which resulted in grouping of sliverlike triangles. In the former case, square-shaped clusters were obtained, while in the latter case, the shape of the clusters corresponds to elongated rectangles. We compared the accuracy of the lighting simulation, assuming that the same numbers of triangles and clusters were used in both cases. As expected, the accuracy for poorly shaped triangles and clusters was significantly lower, and errors increased by 1%–3% in experiments that we conducted for the mesh composed of 19 200 triangles and for various cluster sizes.

4.1.2 Empty sphere with mirrors

The shooting iteration and clustering may be also performed for “virtual lights” used by the “image method” (Sillion and Puech 1994) to model mirror reflections within a radiosity framework. In the experimental validation of our algorithm, we compared simulation and theoretical results for a scene with noticeable influence of specular-diffuse and diffuse-specular-diffuse light propagation mechanisms (Cohen and Wallace 1993). For this purpose, we used the interior of a diffusive sphere with one or two orthogonal mirrors inside (so the real scene is 1/2, or 1/4, of the sphere surface limited by one or two planar mirrors, which pass through the sphere center). One uniform point light source was placed inside to illuminate the scene. This scene was chosen because the exact analytical solution of the light distribution is available (Myszkowski et al. 1994).

The scene we used has the following parameters:

- The sphere has a unit radius and its center is located at the origin of the coordinate system.
- The sphere surface has a diffuse reflectivity $K_d = 0.7$.
- The mirrors’ specular reflectivities are $K_{s1} = 1.0$ and $K_{s2} = 0.8$. The scene SPH2 (half sphere $z > 0$) contains only the first of the two mirrors (with $K_{s1} = 1.0$) in the xy plane, and the scene SPH4 (quarter sphere $z > 0$ and $y > 0$) contains both mirrors (in the xy and xz planes).
- The point light source has an intensity of 100 cd and is placed at the point (0.2, 0.4, 0.6).

We chose three sample points $A = (0, 0, 1)$, $B = (0, 1/\sqrt{2}, 1/\sqrt{2})$, and $C = (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})$ on the sphere surface to compare the theoretical and simulated results. The theoretical results and simulation errors are summarized Table 4.

The solution with clusters (the average cluster size was 6.3 triangles) was almost eight times faster than the regular PR solution. The errors result partially from inaccuracy of the sphere representation as a

Table 4. Results of accuracy analysis for scene of empty sphere with mirror(s)

Scene	Sample point	Theory: illumination [lx]	Experiment: relative error [%]	
			No clusters	Clusters
SPH2	A	686.746	−1.3	−1.9
	B	1038.472	1.3	2.1
	C	939.904	−2.3	−1.2
SPH4	B	1299.769	5.7	6.5
	C	1210.660	1.6	2.2

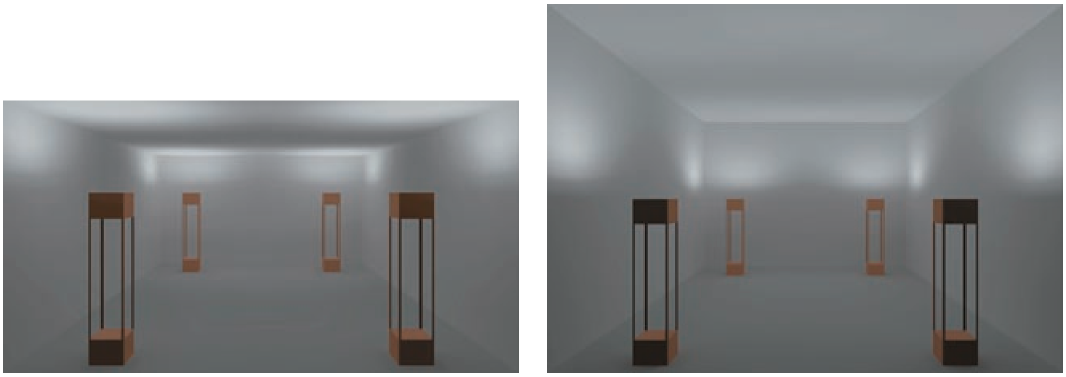


Fig. 6. Computer images of the measurement room for two different ceiling heights

mesh of triangular patches. We used a sphere triangulation with 768 triangles. The maximal difference of length between the sphere radius and a segment connecting the center of the sphere and the nearest point on triangulated surface is 1.5%; this causes 3%, or even more, direct illumination differences, depending on the position relative to the light source. Other known sources of error are approximations introduced into the form factor estimates.

4.2 Comparison with measurements results

Our algorithm was also experimentally validated for more complex scenes by comparing simulation results with measured real-world data. To our knowledge, such a comparison has been done only for the conventional radiosity algorithm (Meyer et al. 1986), while the PR and HR algorithms have not been validated experimentally. Our tests were performed for two scenes. The only difference between the scenes is a different height of the measurement rooms (Fig. 6). Each scene is illuminated by four light sources exhibiting various spatial distributions of candle power. The light sources are attached to movable stands and directed toward the ceiling so that the floor is not directly illuminated. The measurement was performed for 35 points on the floor. Simulation results (indirect light only) for the scenes in Fig. 6 are summarized in Table 5. The scene model was built of over 4000 initial mesh patches. The highest simulation errors were observed at the measurement points located near the corners and along the edges between the floor and the walls. The best results were obtained near the center of the room, in which region the error values were usually less than 2%. To reduce the influence of the discretization error on the simulation accuracy (refer to Sect. 4.1.1) we performed the energy gather step based on our radiosity solution for small differential regions centered at the measurement point locations.

4.3 Comparison with a real-world photograph

The experiments discussed so far were designed specifically to test the accuracy of lighting simulation. However, in many applications, what really matters is the quality of rendering as perceived by the human observer. We designed an additional experiment to investigate this issue, in which we simultaneously viewed synthetic images and photographic

Table 5. Clustering vs. simulation error and calculation time for the measurement room scene

Scene	Average cluster size	Average error (percentage)	Maximum error (percentage)	Time (min:s)
Low ceiling	No clusters	3.9	8.8	2:57
	4.7 triangles	4.3	9.9	0:41
	9.8 triangles	5.1	10.4	0:13
High ceiling	No clusters	3.7	9.6	3:00
	4.3 triangles	4.0	10.8	0:45
	9.3 triangles	4.6	11.4	0:14

images. As a case study, we have chosen an atrium at the University of Aizu (Fig. 7a). In our work on the atrium, we began by specifying the light sources and surfaces in our model according to information from luminaire manufacturers and architectural drawings. We set the reflectance properties of materials in the atrium on the basis of our experience and data available in the literature for similar materials. More complete measurement of reflectance functions would be desirable, but in practice, it was very difficult to obtain comprehensive and reliable data.

In the lighting simulation phase, we obtained luminance and chroma values for every pixel of the final image (the chroma were maintained separately to avoid color shifts). We then transformed the stimulus luminance values to brightness values [perceived brightness predicted for the observer with Stevens' power law (Tumblin and Rushmeier 1993)]. The brightness was also transformed for the range of luminances produced by the display device. First, we confirmed that we were able to present the photographic image under display conditions that gave the best match between the image and original environment. For this purpose, we manipulated brightness, contrast, and the gamma correction of the photographic image. Next, the parameters of the brightness transformation were adjusted until the appearance of the synthetic image (Fig. 7b) best matches that of the photographic image (Fig. 7a). Since we were unable to determine directly a given subject's light adaptation level, this method allowed the brightness function to be adjusted to that subject and given viewing conditions.

Though our approach was rather simple, it proved to be useful in improving the quality of the result produced by our lighting simulation software. While our atrium rendering is still far from perfect, first impressions have been very favorable. In fact, many viewers who were quite familiar with the real atrium thought that they were viewing the actual photographic images when they first viewed the synthetic images. This means that, in terms of absolute evaluation, the quality of our images was acceptable. However, when the same viewers compared the synthetic images to the photographic images, they were able to find many differences and therefore were able to provide us very useful feedback on our atrium model inaccuracies as well as our rendering algorithm deficiencies.

We hope that the global illumination community will come to a more standardized approach to evaluating their algorithms and software. Towards this end, we have posted our atrium data in various formats (The Atrium web page 1997).

The atrium model was originally composed of about 28 200 triangles. The initial mesh is built of 48 700 patches, which are further subdivided into 505 100 triangles. The atrium scene is illuminated by 108 light sources. Our hierarchical radiosity with the shooting iteration required 3 h 11 min, and ray tracing using results of indirect lighting simulation took 4 h 27 min to generate the antialiased and textured image (at a resolution of 1280 × 1024).

5 Results

The results of lighting simulation produced by our algorithm can be used for walkthrough animation and still-image generation. In the former case, we use a texture-mapping technique to store shading for regions where illumination is extremely complex in order to eliminate many deeply subdivided mesh patches and speed up rendering (Myszkowski and Kunii 1994). Ray tracing is used to calculate high-quality images with the results of secondary lighting simulation. Primary lights are recalculated in order to upgrade the image quality.

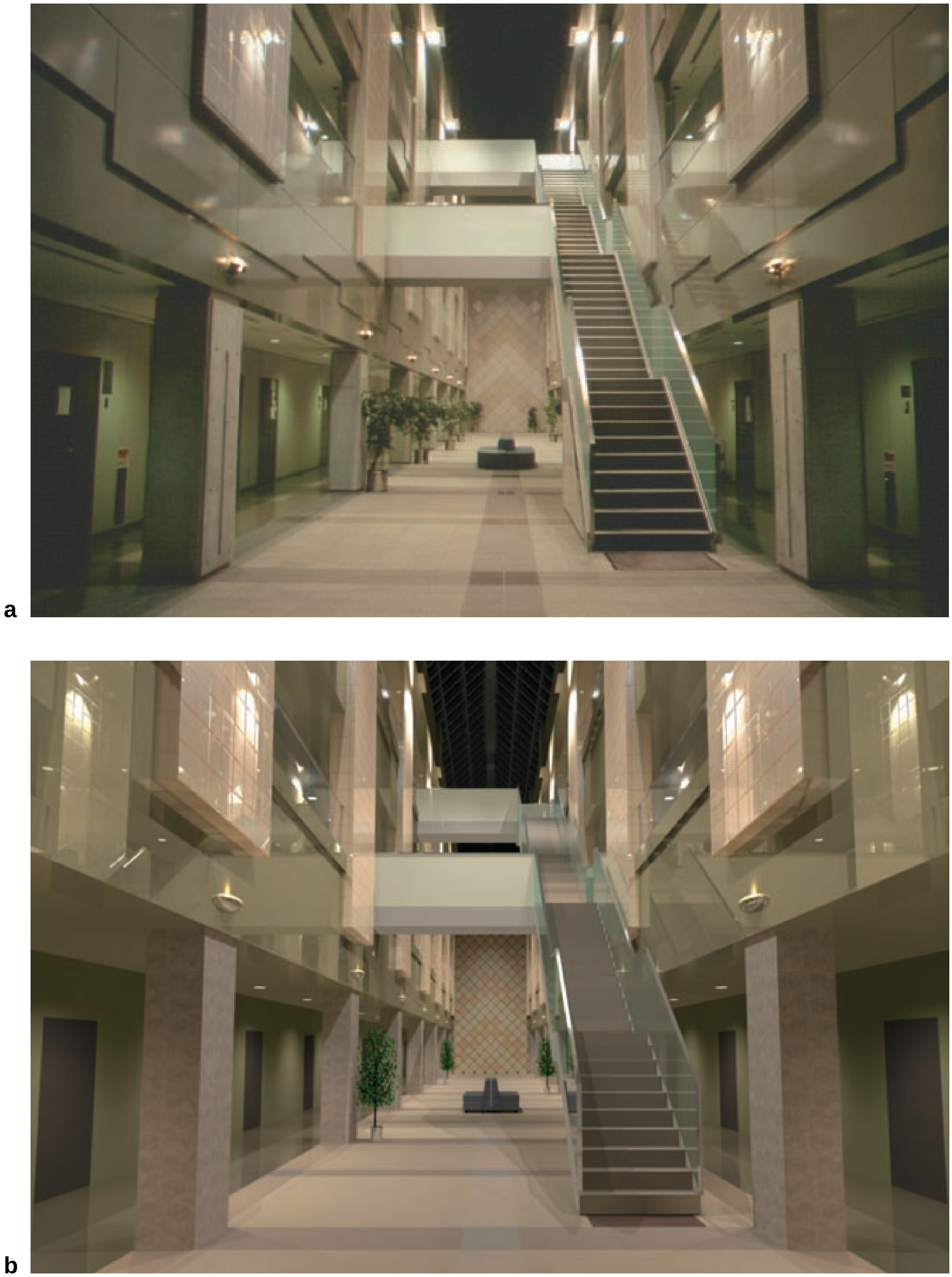
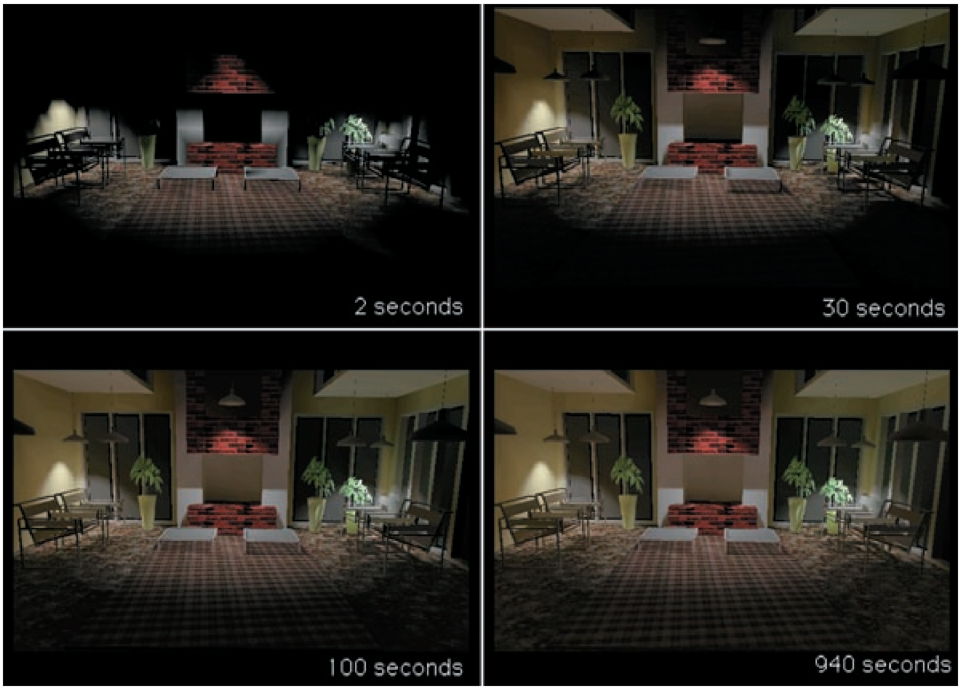
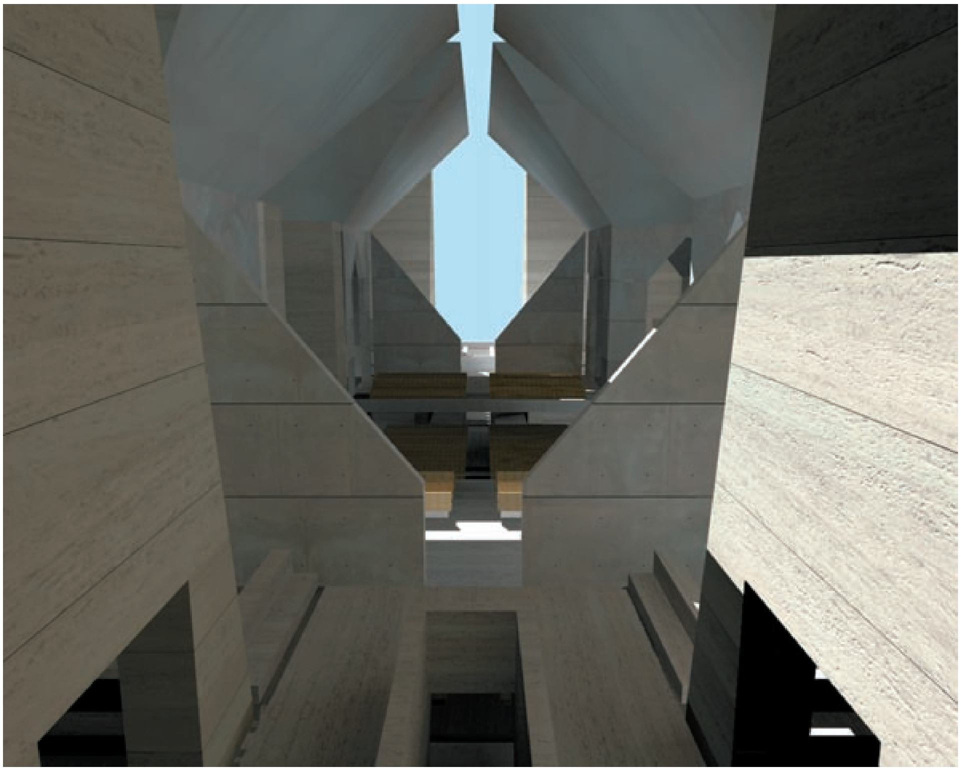


Fig. 7a,b. An atrium of the research quadrangle at the University of Aizu: **a** photograph; **b** rendering



8



9

Fig. 8. Four stages of progressive refinement of the radiosity solution
Fig. 9. The Hurva Synagogue

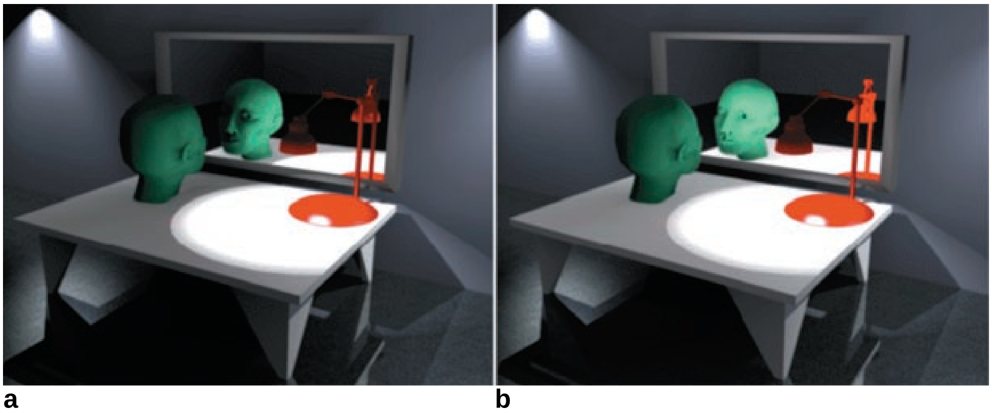


Fig. 10a,b. Clustering of “virtual lights” for the “image method”: **a** the mirror was ignored; **b** it was considered during the shooting iteration. Note the complex light path that was simulated in the latter case: the face is mostly illuminated by light emitted by the red lamp, which is reflected by the table and then by the mirror

Figure 8 illustrates the progressive refinement of the image quality resulting from our algorithm. Successive stages of the radiosity solution are shown for a scene composed of about 5000 triangles. The initial mesh subdivision resulted in 8646 patches, which were grouped into 1261 clusters. The maximal number of initial patches assigned to a cluster was 26. The adaptive mesh subdivision was done only in the regions illuminated by primary light sources, and resulted in 59 104 patches. Time-consuming subdivision of patches into elements due to secondary lights was not performed, while the quality of shading remains reasonable (the same strategy was applied for the scene in Fig. 9).

We checked our algorithm performance for more complex models as well, such as the unbuilt, modernist Hurva Synagogue designed by the famous American architect Louis I Kahn (Fig. 9). The interior of the synagogue is illuminated solely by sunlight, which leaks into the building through narrow openings in the ceiling and gaps between the stone pylons. The scene model is built of 38 429 initial mesh patches. Our radiosity software required 3 h 24 min to simulate the lighting for the test scene.

Extension of the radiosity algorithm to handle mirror reflections for planar mirrors is quite straightforward. Usually the “image method” is applied for this purpose (Sillion and Puech 1994), which requires computation of “virtual lights” for every source patch illuminating the scene via mirror reflection. Without clustering these patches, this solution would be intractable for many practical scenes. Figure 10 illustrates the specular-to-diffuse energy transfer for secondary Lambertian emitters for which only our clustering technique is used. Spotlights are directed toward the floor and the surface of the table in such a way that the mirror on the wall is not directly illuminated. Figure 10a shows results of simulating purely diffuse interreflection (calculation time about 13 min). In Fig. 10b, the head is significantly brighter because indirect light reflected by the mirror was also considered. Clustering of the secondary emitters and shooting iteration were applied to “virtual lights,” which resulted in a moderate increase of the overall computation time to 16 min.

Since our algorithm exhibits only moderate requirements of memory that grow linearly with the number of patches (all data structures needed for light interactions between patches and clusters are created on the fly), we were able to perform all the experiments discussed in this paper on a machine equipped with 64 MB of memory.

6 Conclusions

We have presented a cluster-based hierarchical approach to secondary lighting simulation within the framework of a progressive radiosity algorithm. Although all light interactions are based on the shooting iteration, the hierarchy refinement always involves the total energy transferred (as in the gathering iteration approach), which improves the solution accuracy. The clusters are built automatically upon premeshed scene models, which improves their locality and decreases the dependency on input geometry.

The overall accuracy of the lighting simulation and the resulting image quality provided by our algorithm were validated in a number of diverse experiments. The algorithm performs well for complex scenes, and significantly reduces the calculation time compared to traditional PR algorithms, while progressivity and low-memory requirements are not compromised.

A drawback of the algorithm is the manual setting of several control parameters, including the initial mesh density and the size of the clusters, which requires human expertise. Making these settings fully automatic is an important aim of our future research. Another future research goal of high importance is to establish an experimental framework for the global illumination algorithms testing and comparison. The proposed set of validation tests in this paper is our initial step toward this goal.

Acknowledgements. Turbo Beam Tracing software developed by Integra, was used by the authors as a testbed for implementing cluster-based radiosity. Analytical solutions for the empty cube test were provided by Andrei Khodulev. The model of the atrium was prepared by Edward A. Kopylov, Kouji Honma, Yuya Yaguchi, Seiji Ikuta, Toshiharu Seya, and Toshitake Mashiko. The model of the Hurva Synagogue was provided by Kent Larson and Koji Tsuchiya.

The authors would like to thank Przemek Rokita and Michael Cohen of the University of Aizu for reviewing the manuscript. Special thanks for helpful discussions go to Akira Fujimoto.

References

1. Christensen PH, Lischinski D, Stollnitz EJ, Salesin DH (1997) Clustering for glossy global illumination. ACM Trans Graph 16:3–33
2. Cohen MF, Wallace JR (1993) Radiosity and realistic image synthesis. Academic Press Professional, London
3. Cohen MF, Greenberg DP, Immel DS, Brock PJ (1986) An efficient radiosity approach for realistic image synthesis. IEEE Comput Graph Appl 6:26–35
4. Cohen MF, Chen SE, Wallace JR, Greenberg DP (1988) A progressive refinement approach to fast radiosity image generation. (ACM SIGGRAPH ’88 Proceedings) Comput Graph 22:75–84
5. Gibson S, Hubbold RJ (1996) Efficient hierarchical refinement and clustering for radiosity in complex environments. Comput Graph Forum 15:297–310
6. Hanrahan P, Salzman D, Aupperle L (1991) A rapid hierarchical radiosity algorithm. (ACM SIGGRAPH ’91 Proceedings) Comput Graph 25:197–206
7. Holzschuch N, Sillion FX, Dretakkis G (1994) An efficient progressive refinement strategy for hierarchical radiosity. Proceedings of the 5th Eurographics Workshop on Rendering, Darmstadt, Springer, Berlin, Heidelberg, New York, pp 343–357
8. Kok A (1993) Grouping of patches in progressive radiosity, Proceedings of the 4th Eurographics Workshop on Rendering, Paris, pp 221–232
9. Lischinski D, Tampieri F, Greenberg DP (1993) Combining hierarchical radiosity and discontinuity meshing. (ACM SIGGRAPH ’93 Proceedings) Computer Graphics Annual Conference Series, pp 199–208
10. Meyer GW, Rushmeier HE, Cohen MF, Greenberg DP, Torrance KE (1986) An experimental evaluation of computer graphics imagery. ACM Trans Graph 5:30–50
11. Myszkowski K, Kunii TL (1994) Texture mapping as an alternative for meshing during walkthrough animation. Proceedings of the 5th Eurographics Workshop on Rendering, Darmstadt, Springer, Berlin, Heidelberg, New York, pp 375–388
12. Myszkowski K, Kunii TL (1995) An efficient cluster-based hierarchical progressive radiosity algorithm. ICSC ’95, Hongkong (Lecture Notes in Computer Science, Vol 1024), Springer, Berlin, Heidelberg, NewYork, pp 292–303
13. Myszkowski K, Wicynski K, Khodulev A (1994) Simulation of ideal specular light path by ray tracing. Machine Graph Vision 3:123–137
14. Rushmeier HE, Veerasamy A, Patterson C (1993) Geometric simplification for indirect illumination calculations. Proceedings of Graphics Interface ’93, pp 227–236
15. Sillion FX (1995) A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. IEEE Trans Visualization Comput Graph 1:240–254
16. Sillion FX, Drettakis G (1995) Feature-based control of visibility error: A multiresolution clustering algorithm for global illumination. (ACM SIGGRAPH ’95 Proceedings) Computer Graphics Annual Conference Series, pp 145–152
17. Sillion FX, Puech C (1994) Radiosity and Global Illumination. Morgan Kaufmann, San Francisco
18. Smits BE, Arvo JR, Salesin DH (1992) An importance-driven radiosity algorithm. (ACM SIGGRAPH ’92 Proceedings) Computer Graphics Annual Conference Series, pp 273–282
19. Smits BE, Arvo JR, Greenberg DP (1994) A clustering algorithm for radiosity in complex environments. (ACM SIGGRAPH ’94 Proceedings) Computer Graphics Annual Conference Series, pp 435–442
20. Stamminger M, Schirmacher H, Slusallek P, Seidel HP (1998) Getting rid of links in hierarchical radiosity. (Eurographics ’98) Comput Graph Forum 17:165–174
21. Teller S, Fowler C, Funkhouser T, Hanrahan P (1994) Partitioning and ordering large radiosity computations. (ACM SIGGRAPH ’94 Proceedings) Computer Graphics Annual Conference Series, pp 443–450
22. The Atrium web page (1997) `http://www.u-aizu.ac.jp/labs/csel/atrium`
23. Tumblin J, Rushmeier HE (1993) Tone reproduction for realistic images. IEEE Comput Graph Appl 13:42–48
24. Willmott A, Heckbert P (1997) An empirical comparison of progressive and wavelet radiosity. Proceedings of the 8th Eurographics Workshop on Rendering, St. Etienne, Springer, Wien, New York, pp 175–186

KAROL MYSZKOWSKI is an Associate Professor in the Department of Computer Software at the University of Aizu, Japan. He received his PhD degree in Computer Science from Warsaw University of Technology, Poland in 1991. He was a research scientist at the Technical University of Szczecin, Poland, and a Senior Engineer at Integra, Japan, developing global illumination and rendering software. His current research is investigating the role of human perception as an aid to improving the performance of photorealistic rendering and animation techniques.

TOSIYASU L. KUNII is currently a Professor at Hosei University, an Honorary Visiting Professor of University of Bradford, Professor Emeritus of the University of Tokyo, Adviser of Fukushima Prefecture in Science and Technology, and Liaison Counselor of Fukushima Prefecture Industrial Technology Foundation. He received his BSc in 1962, MSc in 1964, and DSc in 1967, all from the University of Tokyo. He received the 1998 Taylor L. Booth Education Award of the IEEE Computer Society “for initiating and promoting computer and information science education in Japan and for seminal contributions towards the integration of computer-based education in all academic disciplines” on November 13 1999.