EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (*2002*), *Number 3*

Grid Based Final Gather for Radiosity on Complex Clustered Scenes

Annette Scheel[†], Marc Stamminger[‡], and Hans-Peter Seidel[†]

[†] Max-Planck Institut für Informatik, Germany
[‡] REVES - INRIA Sophia Antipolis, France

Abstract

Radiosity methods handle large scenes and complex objects using clustering techniques. To reconstruct a high quality image, usually a second very time consuming final gather pass is applied which exactly recomputes the last light transport before reaching the eye. We propose a new final gather technique which is especially suited for scenes with fine polygonal geometry. In such scenes, substantial parts of the incident illumination vary only smoothly across the surfaces and can be reconstructed on a much coarser structure. We therefore propose a final gather reconstruction based on an object-independent 3D grid. The illumination of each sender is investigated separately: If it varies smoothly across a grid cell, it is interpolated between the vertices of the grid cell, or recomputed exactly, otherwise. We further reduce the number of required samples using view-dependent optimizations. So complex objects with a very detailed structure—plants are good example here—exhibit strong masking effects, which can be exploited by our method. Finally, the estimation of penumbra screen sizes can be used to further reduce costly visibility reevaluations.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Copmuter Graphics]: Picture/Image generation I.3.7 [Copmuter Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Hierarchical Radiosity[1] and its derivates are meanwhile established in industry like the automotive industry or architecture because the result can be visualized interactively and is therefore well suited for Virtual Reality demonstrations. Furthermore, the lack of noise as produced by stochastic Monte-Carlo approaches, is considered a big advantage. An interesting discussion from the view of an architect can be found at `www.cgarchitect.com/upclose/ /article1_KL.asp`.

Clustering techniques[17, 21] make radiosity applicable even to very complex scenes. However, in particular with clustering, radiosity solutions usually still contain artifacts which are not acceptable if high quality images for example for an animation are needed. Shadows are hard to represent in the triangle mesh and the hierarchical representation of the illumination leads to artifacts. Additionally, for scenes with industrial complexity, often only a rough radiosity solution can be produced because of the high computational effort and memory consumption.

A technique to reconstruct high quality images from a radiosity solution is called Final Gather. In a second view-dependent step, a ray tracing pass is performed, which recomputes the radiosity for each visible point. The classical Final Gather technique reevaluates the illumination for each pixel by recomputing the area-to-point formfactor and the visibility term for all links arriving at the corresponding patch. This step is extremely expensive, typically several hundred or even thousands of visibility rays need to be evaluated per pixel to obtain high quality results without visible noise.

In this paper we propose a Final Gather approach which is designed for scenes with complex objects, such as fine triangle meshes representing a curved surface, or complex unconnected geometry such as plants. For these objects, the detailed subdivision is too fine to represent illumination; a large fraction of the light exchange happens at cluster level, above the single triangular patches. In order to avoid storing information at patch level in this case, our approach is based on an object-independent 3D grid covering the view frustum.

For single grid points we compute from the radiosity links a list of senders contributing to the illumination in the neighborhood of the grid point. If a sender does not produce strong variations of illumination near the grid vertex, its contribution is interpolated between neighboring vertices, otherwise it is resampled during the Final Gather.

Furthermore, we integrate view-dependent optimizations for very complex scenes. So we account for visual masking during the Final Gather to save computational effort in image areas where textures or other detail reduce the visual sensitivity. Such masking effects become interesting when detailed, high-contrast textures are used or for complex unconnected geometry such as the foliage of trees. Finally, we present a novel view-dependent criterion to decrease sample numbers for penumbra computations.

2. Previous Work

Despite the fact that high quality images from radiosity solutions usually require a further Final Gather pass, the number of publications on this topic is relatively small. A first resampling approach based on a non-hierarchical radiosity solution has been described in [11]. In the context of hierarchical radiosity, Final Gather was first described in [12] (called 'local pass'), and shortly after in [20]. Using radiosity results to generate an importance function for a path tracer has been proposed in variations in [2, 14, 22]. In [4], a Final Gather step for glossy radiosity has been presented. Finally, [13] proposes a final gather step in the context of bidirectional texture functions (BTFs).

All these approaches reduce the number of links to be resampled. However, they suffer from continuity problems along patch boundaries. When a sender is resampled for one patch, but not for its neighbor, a discontinuity is likely to appear. Even if this discontinuity is very small, it can be well perceived by an observer. This deficiency has been lifted by [16], where importance distributions are continuously interpolated over patches, however, as the authors state, this interpolation on patch level becomes inefficient for complex geometry. The method presented in this paper borrows ideas of [16], however the use of an object-independent 3D-grid decouples its efficiency from scene complexity. In this context, Photon Maps [10] and Irradiance Volumes [8] are related, because they store and interpolate illumination in object-independent spatial data structures. A 3D grid which is placed into the bounding box of complex objects has been used by [5] for interpolation of so called macro-scale visibility (shadows due to far away blockers) between grid vertices.

The idea of exploiting properties of human visual perception has been introduced in [3] and [15]. In their work models for human vision were developed which are used to guide a Monte-Carlo based global illumination solution, since Monte-Carlo approaches allowed them a simple, spatially varying accuracy control.

3. Motivation

Radiosity methods handle complexity of scenes by hierarchical refinement which subdivides existing patches, and by clustering which completes the hierarchy 'above' the patch level. Complex objects are grouped to a cluster or a hierarchy of clusters. A lot of computation time as well as memory is saved if the light transport is approximated by the transport between clusters.

Clustering makes it possible to compute radiosity solutions even for complex scenes, however it also introduces problems. For complex scenes, the majority of links is not refined to patch level but arrives at clusters. These are mainly links carrying indirect light but also links from distant light sources. However, the formfactor and the visibility information in the link is very imprecise if the cluster contains many objects or covers a wide area. Often the visibility is only determined outside the cluster, visibility inside the cluster is just approximated by a coefficient [19, 18]. The self-link of a cluster, i.e. the light exchange within a cluster, is also approximated very roughly. Furthermore, the precision of the links and hence the quality of the radiosity solution depends on the quality of the clustering hierarchy. Clustering algorithms which automatically create a 'good' clustering hierarchy are hard to design[9]. Face Clustering [24], which groups triangles within a faceset that have similar normals, introduces a better hierarchy on widely connected faceset. However, for objects with a 'fuzzy' structure like plants this method is less beneficial, because continuous areas with similar normals are very small. To summarize, we can say that the results of a radiosity clustering algorithm usually do not meet high quality requirements, so a costly Final Gather step is almost mandatory.

Optimized Final Gather approaches require interpolation of illumination. One solution is to perform the Final Gather at patch level. In [16], it is proposed to push down all links to the triangle level for all visible triangles. For the pushed down links, the formfactor is recomputed (which is necessary to obtain a high quality interpolation), but not the visibility estimate. This is also possible with clustering, but because visibility is approximated very roughly inside a cluster, a recomputation of visibility becomes necessary as well. Also the self-link, which would resolve into thousands of links, imposes problems. The push down of links is not only time but also memory consuming, even if finally only links which need exact resampling are stored which is a small fraction of the total number of links. Additionally, the push down of links to triangle level is often unnecessary as well. As stated in [24], for illumination often a much coarser triangle mesh is sufficient than it is need for geometry. For detailed geometry, the contribution of most senders can be smooth over a wide range of triangles. The triangle mesh is then not suited to represent the illumination.

Instead, we decided to base the Final Gather step on a three dimensional grid covering the visible part of the scene.

The approach is similar in spirit to [16], but the resampling information is not gathered at the patch vertices, but at the grid vertices. So for each grid vertex we compute a slowly varying irradiance component, which can be safely interpolated within the grid cells, and a list of senders that result in a strongly varying illumination and thus require resampling.

4. Algorithm

4.1. Overview

The proposed algorithm describes a method for rendering a high quality image from a hierarchical radiosity solution with clustering. Rays are traced from the camera through the pixels of the image until they hit the first object. We check for each link arriving at the hit object, if the contribution of the links' sender is smooth and can thus be interpolated or really requires resampling. For this purpose a 3D grid is established in the scene. At the vertices of the grid a list of all senders is built using the links arriving at the cluster(s) around the grid vertex. Each sender contribution is examined and the sender is classified either as to be resampled or as to be interpolated within the grid. After the classification we obtain a (small) list of senders with significant contributions or which cause visibility problems.

For the uncritical senders, we evaluate the irradiance exactly at the grid vertices and interpolate these values between neighboring vertices during Final Gather. For the senders to be resampled, an importance-driven sample distribution is established for each vertex. For a point inside a grid, we interpolate the sample numbers of the grid vertices per sender in order to obtain a continuous sample distribution across all grid cells, just as it is done in [16].

The method described here is particularly suited for reconstruction of links arriving at cluster level. Links which have been refined to the patch level by the first radiosity pass are better processed by a Final Gather which operates on surfaces directly. We will describe in Section 4.5, how we combine these two approaches.

4.2. Grid

The new Final Gather technique requires a 3D grid in the scene. We use a non-hierarchical, *perspective* grid, where the grid cell size varies with the distance to the camera. Nearby grid cells are small, distant cells are larger. We obtain this grid by generating a uniform grid in post-perspective space (i.e. the 3D space after the perspective transformation has been performed), and applying the inverse perspective and camera transformation. Due to this construction, the grid just covers the viewing frustum; each grid cell is a truncated pyramid (see Figure 1).

The advantage of this perspective grid is that the grid distance is rather constant in screen space. Thus, for nearby objects the classification step is performed on a finer grid (in object space) which decreases the risk of missing fine detail. The method can of course work with a regular, view-independent grid as well.

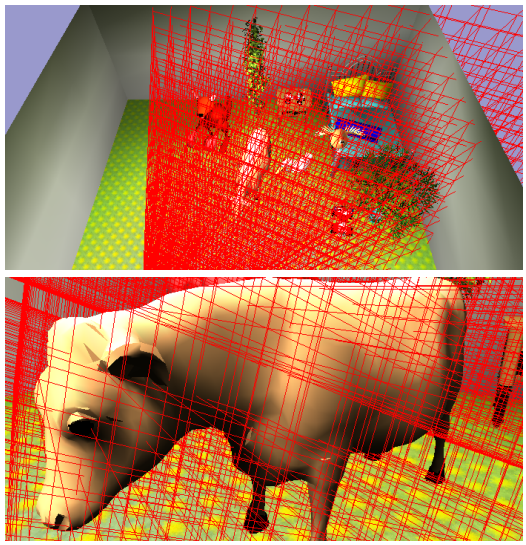

Figure 1: *Perspective grid, seen from above and from nearby. The camera viewpoint is at the apex of the grid pyramid.*

We also considered using a hierarchical grid, and refining grid cells in areas with strongly changing illumination. Similar to a triangle mesh we then need to insert 'T-vertices' into the grid to make a continuous interpolation possible—which results in significant overhead for the bookkeeping of the resulting grid. Furthermore, for objects with a complex structure (e.g., a tree), extremely fine refinement of grid cells is necessary to capture most of the illumination details and thus reduce the resampling. This results in a higher storage consumption and an increased classification time which quickly outweighs the benefit.

4.3. Sender Classification

In order to reconstruct illumination at a scene point, we have to find the grid vertices of the surrounding cell. For each of these vertices, the sender list has to bee computed, if this information is not stored in the grid vertex yet.

To compute the sender list, we start with the links which arrive at the clusters in the neighborhood of the grid vertex. If the radiosity system uses a hierarchy of clusters, also links from clusters in the hierarchy above are gathered. Each such link stores a formfactor and visibility estimate which are used to determine an importance distribution for sample numbers. For this we collect the set of senders $s \in S$. The number of samples N_s for a sender s is then set to:

$$N_s = N I_s / \sum_{s \in S} I_s, \qquad (1)$$

where N is the total number of samples and I_s is the unoccluded irradiance as stored in the links. We use unoccluded irradiance, because partially occluded senders should not receive less samples. We skip senders, which are completely occluded according to the link information. For non-conservative visibility classification during the radiosity pass, this involves the risk of missing a sender which was wrongly classified as occluded.

Next, we determine a separate approximation of the formfactor and the visibility term to each sender as well as an estimate of their variation. We choose N_s sample points inside the grid cells surrounding the vertex and on each sender, and compute the formfactor kernel and the visibility term using these samples. The average of these values gives the approximation of the formfactor and the visibility term. The error estimate of the formfactor D_s^F for sender s is obtained by:

$$D_s^F = \rho B_s (F_{\max} - F_{\min}) \mathrm{vis}, \tag{2}$$

where B_s is the radiosity of the sender, ρ the average reflectance at the receivers, $F_{\max/\min}$ the maximal/minimal formfactor and vis the visibility term. Since we compute this information for a region around the grid cell, we do not know the receiver's normal. Thus, $F_{\max/\min}$ do not contain the cosine at the receiver and are thus in fact the solid angle of the sender. The cosine at the receiver will then be computed on the fly.

Similar, the error estimate for visibility D_s^{vis} is:

$$D_s^{vis} = \begin{cases} 0, & \mathrm{vis} = 1 \\ B_s F_{\mathrm{avg}}, & 0 < \mathrm{vis} < 1 \\ 0, & \mathrm{vis} = 0 \end{cases} \tag{3}$$

By computing visibility from random points around the grid vertex, a problem arises: suppose a planar surface is the only object near the grid vertex and it is fully visible to the sender (compare Fig. 2). If we randomly select nearby sample positions, the object is hit by a fraction of the visibility rays. If we interpret these rays as occluded, we obtain a wrong visibility classification. Note that this problem would not exist if samples are always spawned from a surface, but finding a random surface point in a grid cell is costly. Our solution is to count the number of object hits. Only visibility test rays with more than one hit for double sided or more than two hits for single sided objects are considered as occluded (see Figure 2).

If the sender s is a cluster containing the grid vertex, formfactor and visibility errors can become arbitrarily large. On the other side, the contribution of such clusters cannot be significant or cause significant detail, otherwise the corresponding (self-)link would have been refined. Thus we always interpolate their contribution. We therefore shoot rays from the vertex into random directions to determine the irradiance due to s. This irradiance is then added to the slowly varying irradiance for later interpolation. Again, the number of samples used is not critical, because this operation is not performed often. We usually shoot about 100 samples.

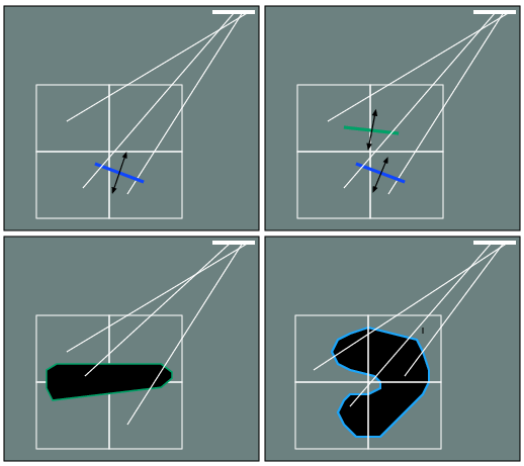

Figure 2: *Visibility computation. Top row: double sided objects: shadows exist, only if a ray hits more than one object. Bottom row: single sided objects: shadows (and self-shadows) exist, only if a ray hits more than two object boundaries.*

The two error estimates, for the formfactor and for the visibility, are used to classify how the illumination due to the sender should be computed. If the error estimates exceed a user specified threshold (Sect. 5 describes how this threshold can be chosen taking visual perception into account), the sender is classified as to be resampled, otherwise its contribution is determined by interpolation between the grid vertices. We completely separate formfactor and visibility; a sender might hence have the following classifications:

- Resample Visibility & Resample Formfactor
- Resample Visibility & Interpolate Formfactor
- Interpolate Visibility & Resample Formfactor
- Interpolate Visibility & Interpolate Formfactor

4.4. Rendering

The result of the classification procedure are two lists for each vertex. The first list contains the set of senders which have to be resampled and the corresponding number of samples. The second list consists of the contribution of senders which can be interpolated. In the second list, we store for each sender the incident radiance and the direction to its center, so we can account for the cosine between the incident direction and the surface normal later. Due to the low number of grid vertices the storage overhead for this list is not significant. Alternatively, it would be possible to store just average illumination values for a predefined set of directions similar to the Irradiance Volumes [8].

For rendering a point we interpolate the sampling infor-

mation of surrounding grid vertices. To reduce the computational effort, the following 'trick' is used: each grid cell can be subdivided into five tetrahedra without inserting new vertices (see Fig. 3). Instead of querying and interpolating between the *eight* vertices of the cell we only use the *four* vertices of the corresponding tetrahedron which surrounds the object.

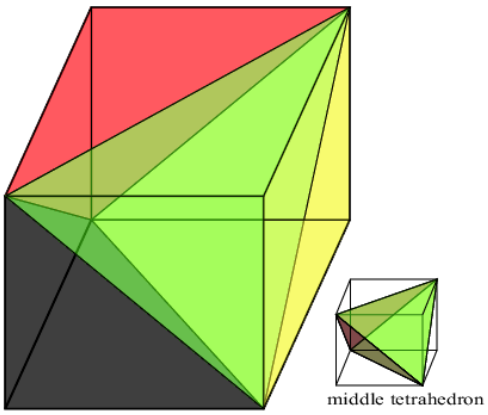

Figure 3: *Subdivision of a grid cell into 5 tetrahedron*

4.4.1. Resampling of Senders

For resampling a particular sender, first the number of samples is determined by interpolating the sample numbers stored at the four vertices depending on the position of the point inside the tetrahedron. This avoids discontinuities at grid boundaries due to discontinuous importance functions. The sample number is then used to recompute the light transport between the sender and the point on the object.

Sample points on the sender are chosen randomly, but once chosen the position of the samples is fixed, i.e. the sample positions do not change from one pixel to the other. This introduces bias, but we found that bias is much less disturbing than the noise which appears when the sample positions are chosen differently for each pixel.

4.4.2. Interpolation

For interpolation of uncritical contributions, we compute the cosines at the receiver for the different irradiance samples and sum up the resulting irradiances for each grid vertex. These irradiance values are then interpolated according to the point's relative position in the tetrahedron.

If one vertex has classified a sender as to be interpolated but another vertex of the tetrahedron classified the same sender as to be resampled we do both and interpolate the two results afterwards. By this, continuity in the solution is maintained and no visible switches at tetrahedron boundaries appear.

4.5. Combination with Surface-based Final Gather

As mentioned previously, the grid based Final Gather is best suited for the reconstruction of illumination arriving at cluster level. Since the objects inside a cluster are not necessarily connected, our spatial, object-independent grid is used as basis for interpolation. For fine hierarchical subdivisions of large initial patches (for example the walls in a room), it is more natural to use the triangle mesh obtained from the subdivision for interpolation on the surface, which is already adapted to lighting detail.

We thus combine grid- and surface-based Final Gather. If a scene object is subdivided to a patch size smaller than the grid distance at the patch, we apply a surface-based Final Gather for this patch. If the smallest subdivision of the object is still coarser than the grid or if it belongs to a cluster which was not refined, grid-based Final Gather is used. This typically is the case for complex clustered scene objects, where the geometric complexity is larger than the complexity of the radiosity solution.

5. Integration of View-Dependent, Perceptual Aspects

In the algorithm described so far the Final Gather was only steered by view-independent information, namely link information from the radiosity pass plus additional formfactor and visibility test rays for classification. However, several aspects which arise from the limitations of the display device on the one hand and from visual perception on the other hand would allow us to reduce the effort for the Final Gather even further without decreasing quality. In particular masking effects, i.e. the reduction of the eye's sensitivity due to concurrent frequencies, have been exploited already in computer graphics [15, 3].

The Final Gather contains two critical factors which directly influence the image quality and the computation time: 1. the determination of critical senders which need resampling and 2. the number of visibility tests for a sender. In the following we will show how the decrease of visual sensitivity due to masking effects can be exploited to reduce the number of critical senders needing resampling. Additionally, we investigate how the number of visibility tests can be adjusted to the current view. We start with a brief overview of factors which influence visual sensitivity.

5.1. Contrast Sensitivity and Visual Masking

Textures or complex structures can hide small imperfections in the reconstruction. Consider for example the left smooth shadow in the first row of Figure 4: it is clearly visible on the uniformly colored floor on the left but hard to recognize on the textured floor on the right. Furthermore, the observer will hardly notice if small shadows in the plant are missed.

The effect described here is known as *visual masking*.

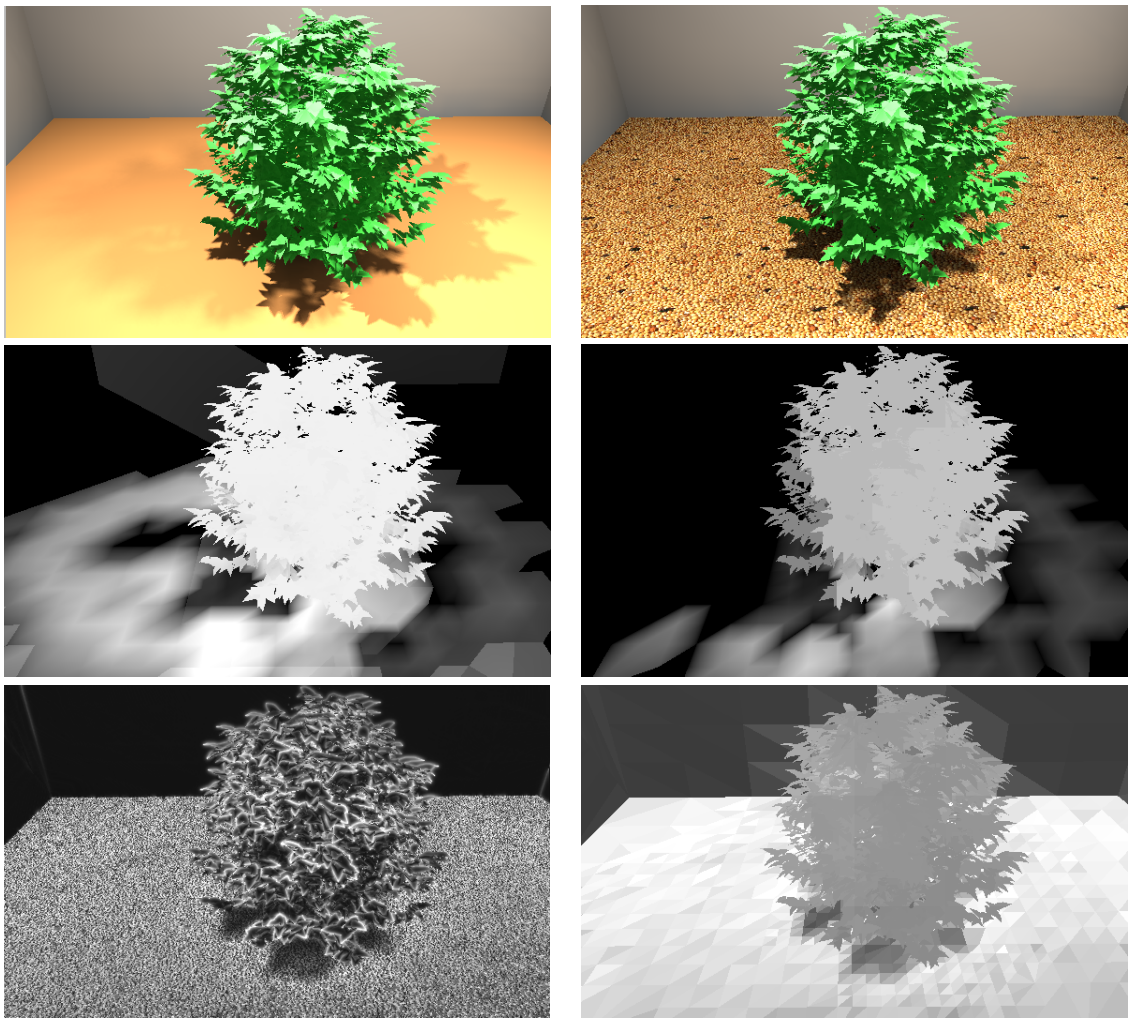

Figure 4: *Masking by textures and complex objects. Top Row: Final gather result without masking (left) and with masking (right). Middle row: Number of visibility samples without masking (left) and with masking (right). Bright regions correspond to high sample numbers. On the textured floor, less shadow detail is visible, so less samples are sufficient. Bottom row: Masking image (left) Masking per patch/grid cell (right). Bright regions correspond to high masking effects or threshold elevation.*

This phenomenon was investigated and described intensively in literature; a nice comprehensive summary was given in [7]. We will therefore only briefly describe the relevant aspects of the human visual system and the masking model and concentrate then on the practical application of the masking model.

First of all, the perception of contrast depends on the adaptation luminance, e.g. the brightness of the background to which the eye is adapted. The *threshold-versus-intensity (TVI) function* describes this relationship. Over a wide range of luminances the threshold increases linearly with background luminance, but for brighter backgrounds the threshold increases non-linearly (see for example [6]).

The ability of the human eye to detect contrasts in patterns differs depending on the spatial frequency and the orientation of the pattern. The sensitivity for spatial frequencies is described by the *Contrast Sensitivity* function, which has a peak at approximately 4–5 cycles/degree and drops off for higher and lower frequencies. Additionally the eye is tuned to particular orientations and again this tuning differs with the spatial frequency of the pattern.

Finally, the presence of one frequency can influence the perception of other frequencies. This effect is known as *visual masking*. The recognition of contrast is most difficult for patterns with similar frequencies and orientations and for high contrasts.

In our method we use the *threshold model* developed by [15] for a similar purpose. Here, the model was used to steer

the indirect light calculation with a path tracing algorithm. They first created an image which contained only direct light which is relatively inexpensive to compute and contains already most of the lighting detail and added an ambient term to account for indirect light. For this image the threshold model determines masking effects due to textures, complex structures and lighting detail. In areas with high masking the number of rays could be significantly reduced, because noise was less visible. Next, we will describe how we apply this threshold model to the Final Gather.

5.2. Application of the Masking Model for Sender Classification

As already indicated one critical point of the Final Gather is how many senders can be classified as suitable for interpolation. So far we used just a user defined threshold for this decision. In this Section we describe how the threshold can be raised according to masking effects without introducing noticeable artifacts.

After the radiosity computation we obtain a simple approximation of the final result by using gouraud shading. The tonemapped image is used as the input into the masking model. It does not contain fine illumination detail like sharp shadows, but nevertheless already has the right illumination level, textures and all object detail.

The output of the masking model is a threshold elevation factor for each image pixel which describes the elevation of the eye's contrast detection threshold for optimal viewing conditions which is given by the TVI function. Therefore we first have to determine the TVI threshold based on the luminances of the tonemapped radiosity solution. We use the TVI data given in [23]. Then the threshold is multiplied with the elevation factor of the masking model.

Recall from the previous section that we get two error estimates, D_s^F and D_s^{vis} for the formfactor and the visibility term, respectively, which estimate the maximum change in radiosity expected across the grid cell. After tonemapping we can directly compare D_s^F and D_s^{vis} against the new threshold.

As described previously the sender classification is not done for each pixel but only per grid vertex, which is then valid for all grid cells surrounding the vertex. Therefore, the masking based threshold is only needed at the vertices. We use the average masking values in the adjacent grid cells as masking threshold for the vertex. Similarly, for the patch based Final Gather we determine one masking threshold per triangle vertex.

In our example from Figure 4 the total number of visibility and formfactor samples was strongly reduced when a masking based threshold was used—without decreasing visual quality of the solution. Table 1 shows the number of visibility and formfactor samples as well as computation times for visibility and formfactor terms for this test scene. The total computation time which includes everything (radiosity, classification, masking computation, and rendering) was nearly halved. The time for the masking computation and modification of the masking threshold is about 5 sec. The middle row of Figure 4 visualizes the number of visibility samples (being far more critical than the number of formfactor samples) without and with masking. Especially in the smooth shadow area on the floor the number of samples is much lower with masking which corresponds well to the perception of the textured scene where the shadow is hardly noticeable.

Method	samples		render time		
	vis	ff	vis	illum	total
Without mask.	76	107	273 s.	10 s.	310 s.
With masking	47	68	158 s.	6 s.	169 s.

Table 1: *Sample numbers and computation times with and without masking.*

For results on more complex scenes please refer to the main results Section 6.

5.2.1. Discussion

Our input image to the masking model contains only the radiosity solution with very blurred shadows, but the shadows that will be added by the Final Gather can have completely different frequencies. Following the theory on masking, only patterns of similar frequencies and orientations influence each other strongly. How strong the shadow-frequencies will influence the sensitivity of the eye can thus not be determined exactly by the model because they are not known in advance. To attenuate this problem, one could try to approximate the most important shadow details beforehand by approximating the strongest area light sources as point lights and using visibility maps. However, our results show that the application of the masking model already works well without this step. Note that the approach of [15] faces a similar problem: They try to estimate in advance if the noise introduced by the path tracer will be masked out by other frequencies of the image. If the image contains mostly different frequencies than the noise the reduction of sensitivity might be lower than expected.

A further critical point of our method is that we need a masking threshold per grid/triangle vertex and not per pixel in order to maintain a continuous reconstruction. In the masking image shown for example in Figure 4 it can be seen that masking can vary rapidly. Ideally, the masking value of an object should be the minimum masking value of all pixels that see the object. In our experiments, we found that this rule is too conservative, so we use the average masking value instead.

5.3. View-dependent Number of Samples for Shadows

The Final Gather uses an importance driven sample distribution. More precisely, given a total number of samples N a sender producing irradiance I_s obtains $N_s = NI_s / \sum_S I_s$ samples. Hence, senders producing high irradiances on the receiver are sampled more densely. For the resampling of the formfactor this criterion achieves good results. The number of samples for visibility is much more critical: on the one hand visibility tests are the by far most expensive component of the Final Gather, so the goal is to use as few shadow rays as possible. On the other hand, due to the strong illumination variations caused by shadows usually a high number of shadow rays is needed to obtain a high quality result.

A view-independent sample number is far from optimal as the top row of Figure 6 shows. It contains two views of the same scene, a close and far one; for the penumbra approximately 150 visibility rays are shot per pixel towards the area light source. Since we select fix samples on the light source, steps become visible in the penumbra, in particular in the left part of the shadow of the close view. When the penumbra is smaller, the sample number is sufficient. So, the larger the penumbra (or the lower the gradient) becomes the more samples are needed to represent it without visible steps.

In general, one can say that the more pixels in the final image a penumbra covers the more samples are needed. The number of pixels covered can be taken as an upper bound on the number of samples, because then the steps become smaller than a pixel and are thus invisible (compare Figure 5). The size of the penumbra is approximated from link information, if the distance to the closest occluder is determined during radiosity computation and stored with the link [16]. Again, this information can only be determined per patch vertex or grid vertex and is then interpolated in order to maintain the continuous reconstruction.

The images of the lower row of Figure 6 were computed with this upper bound. For the close view now about 200 samples per pixel (for the large penumbra areas) are used whereas 50 samples are sufficient for the far view (for the same penumbra area). Additionally the number of samples used changes across the penumbra from 20 (5) for the sharpest part of the penumbra up to a ten times higher value for the most blurry part.

This very simple way of sample number determination does not yet respect the magnitude of change of the radiance inside the penumbra. A darker sender produces a smaller contrast between shadow area and lit area and thus fewer samples are necessary to represent this shadow (compare Figure 5, right). Looking at the penumbra step function, we can say that the steps become invisible if the step height is too small to be visible. This leads to another bound: we use just enough samples that the step difference in the tonemapped result is below the smallest visible contrast. Again, this contrast is adjusted by the elevation factor due to masking. The bottom row of Figure 6 was created combining both criteria; on average, less samples are used compared to the brighter scene, but still the sample number increases towards the bigger penumbra.

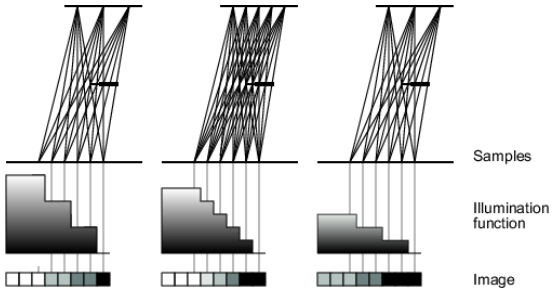

Figure 5: *Penumbra. The fix sample positions on the sender produce a step function on the receiver with approximately as many steps as samples. If the illumination is lower (right) the step height becomes smaller.*

6. Results

Our first test scene shown in Figure 7 contains several complex objects and consists of 250,000 triangles in total. The walls were computed with the surface based Final Gather of [16], but for all other objects Final Gather was computed using the grid based method. Table 2 lists the sample numbers and computation times with and without masking. Without masking on average only 45 samples for visibility computation have been used. The total computation time for the image of size 700 by 350 is 314 seconds including classification, masking computation, and rendering. The radiosity solution took about 7 seconds. By far most time was spent on the visibility computation during rendering. If we apply the masking model, the number of visibility samples decreases to 37, leading to a reduction of the computation time by 60 seconds. Figure 7 shows the resulting image; below we added two renderings of an image detail without textures for demonstration: the top one shows a Final Gather solution on the scene without textures, the bottom one the Final Gather solution from the large image, but with the textures removed. It can be clearly seen how the shadow detail in the lower right is removed due to the masking effect of the floor texture. The small images on the right visualize the visibility sample numbers with and without masking (top and center top). They show that the number of shadow rays is also reduced at the walls behind the plants due to the masking effects of the plants.

The second test scene is rather complex, contains about 80 large area lights (windows and ceiling lights), and consists of about 500,000 triangles (see 8). Only a very rough radiosity solution was computed in 70 seconds. The two lower rows of Table 2 list the timings for image resolution 630x350. With

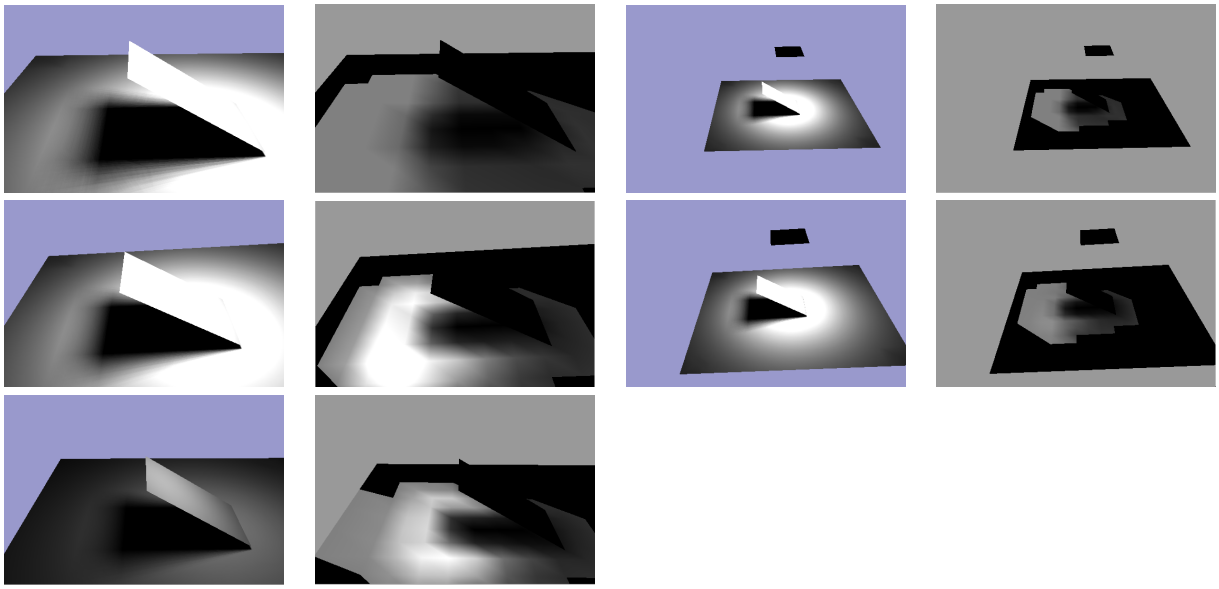

Figure 6: *Visibility Samples Numbers. Top row: Fix total sample number. The same sample numbers are used for the sharp and the blurry part of the penumbra as well as for the close and the far view (bright: high number of samples). Middle row: View-adjusted sample numbers. For the close view the sample numbers vary from 20 to 200, for the far from 5 to 50, for the sharp and the blurry part, respectively. Bottom row: If the sender is darker, fewer samples are needed to represent the penumbra. Here the sample numbers range from 15 to 150.*

masking, only 94 visibility samples were computed on average. The total time needed for the visibility tests was 820 seconds. For both scenes, the grid contained about 10.000 vertices but only half of them were actually used. The efficiency of our BSP acceleration structure was low for that scene, the fine detail in the plants resulted in a large number of scene objects in the tree's leaf nodes. The memory needed for the Final Gather pass was about 100 MB.

Our models have been downloaded from www.3dcafe.com, www.inf.tu-dresden.de/ST2/cg/downloads/publicplants (xfrog models) and textures from www.fortunecity.com. All scenes were computed on a 1.1GHz Pentium Linux PC with 768 Megabytes. We used an adaptive oversampling technique which spawns additional samples at object borders.

Masking	samples			time (sec.)		
	vis	ff	class	vis	illum	total
No	45	168	7	266	25	314
Yes	37	121	7	216	18	252
No	121	206	52	1197	36	1343
Yes	94	162	52	820	28	946

Table 2: *Sample numbers and computation times with and without masking for the scenes in Figure 7 (upper rows) and 8 (lower rows). Adapted oversampling generated about 2.5 samples per pixel.*

7. Conclusions

We presented a Final Gather approach optimized for complex scenes. Based on the observation that the triangle mesh of many objects is much too fine to represent slowly varying irradiance, we reconstruct on an object-independent 3D grid placed into the scene. We classify senders according to their contribution. Senders which cause smooth and low illumination are interpolated across the grid cells, only critical senders are resampled. We furthermore enhance the method by view dependent optimizations, taking into account visual masking of the human visual system, and image sample resolution. A conservative sample number for visibility tests is determined which guarantees a smooth representation of shadows.

With the resulting method we generate high quality Final Gather results with relatively low sample numbers. Compared to a full Final Gather, we obtain the same visual quality with about one third of the number of reevaluations. By using an object-independent grid for our computations, we decouple the required computation resources from scene complexity, such that reconstruction is possible also

for complex geometry. Even if the input radiosity solution is very coarse, good results are obtained from the reconstruction.

8. Acknowledgments

The first author is supported by the ESPRIT Open LTR Project 35772 "SIMULGEN", Simulation of Light for General Environments. The second author was supported by a Marie-Curie Post Doctoral Fellowship.

References

1. Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 155–162, 1993. 1

2. Philippe Bekaert, Philip Dutre, and Yves Willems. Final radiosity gather step using a monte-carlo technique with optimal importance sampling. Technical Report CW275, Katholike Univ. Leuven, 1996. 2

3. Mark Bolin and Gary Meyer. A perceptually based adaptive sampling algorithm. In *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 299–309, 1998. 2, 5

4. Per H. Christensen, Dani Lischinski, Eric Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1):3–33, January 1997. 2

5. J. Dischler, L. Moustefaoui, and D. Ghazanfarpour. Radiosity including complex surfaces and geometric textures using solid irradiance and virtual surfaces. *Computers and Graphics*, 23(4), 1999. 2

6. James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A model of visual adaptation for realistic image synthesis. In Holly Rushmeier, editor, *Computer Graphics (SIGGRAPH '96 Proceedings)*, pages 249–258, August 1996. 6

7. James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley, and Donald P. Greenberg. A model of visual masking for computer graphics. In Turner Whitted, editor, *Computer Graphics (SIGGRAPH '97 Proceedings)*, pages 143–152, August 1997. 6

8. Gene Greger, Peter Shirley, Philip Hubbard, and Donald Greenberg. The irradiance volume. *IEEE Computer Graphics and Applications*, pages 32–43, March/April 1998. 2, 4

9. Jean-Marc Hasenfratz, Cyrille Damez, François X. Sillion, and George Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. *Computer Graphics Forum*, 18(3):221–232, September 1999. 2

10. Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH'98 Conf. Proceedings*, pages 311–320, 1998. 2

11. Arjan J. F. Kok and Frederik W. Jansen. Adaptive sampling of area light sources in ray tracing including diffuse interreflection. *Computer Graphics Forum*, 11(3):289–298, 1992. 2

12. Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics (SIGGRAPH'93 Proceedings)*, pages 199–208, 1993. 2

13. L. Mostefaoui, J.-M. Dischler, and D. Ghazanfarpur. Rendering inhomogeneous surfaces with radiosity. In *Rendering Techniques '99 (Proc. EG Workshop on Rendering)*, pages 283–292. Springer, 1999. 2

14. F. Perez, I. Martin, X. Pueyo, and F.X. Sillion. Acceleration of monte carlo path tracing for general environments. In *Proc. Pacific Graphics 2000*, 2000. 2

15. M. Ramasubramanian, S. Pattanaik, and D. P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *Computer Graphics (SIGGRAPH'99 Proceedings)*, pages 73–82, 1999. 2, 5, 6, 7

16. Annette Scheel, Marc Stamminger, and Hans-Peter Seidel. Thrifty final gather for radiosity. In *Rendering Techniques 2001 (Proc. of the EG Workshop on Rednering)*, pages 1–12. Springer, 2001. 2, 3, 8

17. François Sillion. Clustering and volume scattering for hierarchical radiosity calculations. In *Fifth Eurographics Workshop on Rendering*, pages 105–117, Darmstadt, Germany, June 1994. 1

18. François Sillion, G. Drettakis, and Cyril Soler. A clustering algorithm for radiance calculation in general environments. In *Eurographics Rendering Workshop 1995*. Eurographics, June 1995. 2

19. François X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995. 2

20. Brian Smits. *Efficient Hierarchical Radiosity in Complex Environments*. Ph.d thesis, Cornell University, 1994. 2

21. Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In *Proceedings of SIGGRAPH '94*, pages 435–442. ACM SIGGRAPH, ACM Press, July 1994. 1

22. W. Stürzlinger. Optimized local pass using importance sampling. In *International Conference in Central Europe of Computer Graphics and Visualization '96*, pages 342–348, 1996. 2

23. Greg Ward-Larson, Holly Rushmeier, and C. Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997. 7

24. Andrew J. Willmott, Paul S. Heckbert, and Michael Garland. Face cluster radiosity. In *Rendering Techniques 99 (Proceedings of EG Workshop on Rendering)*, pages 293–304. Springer, 1999. 2

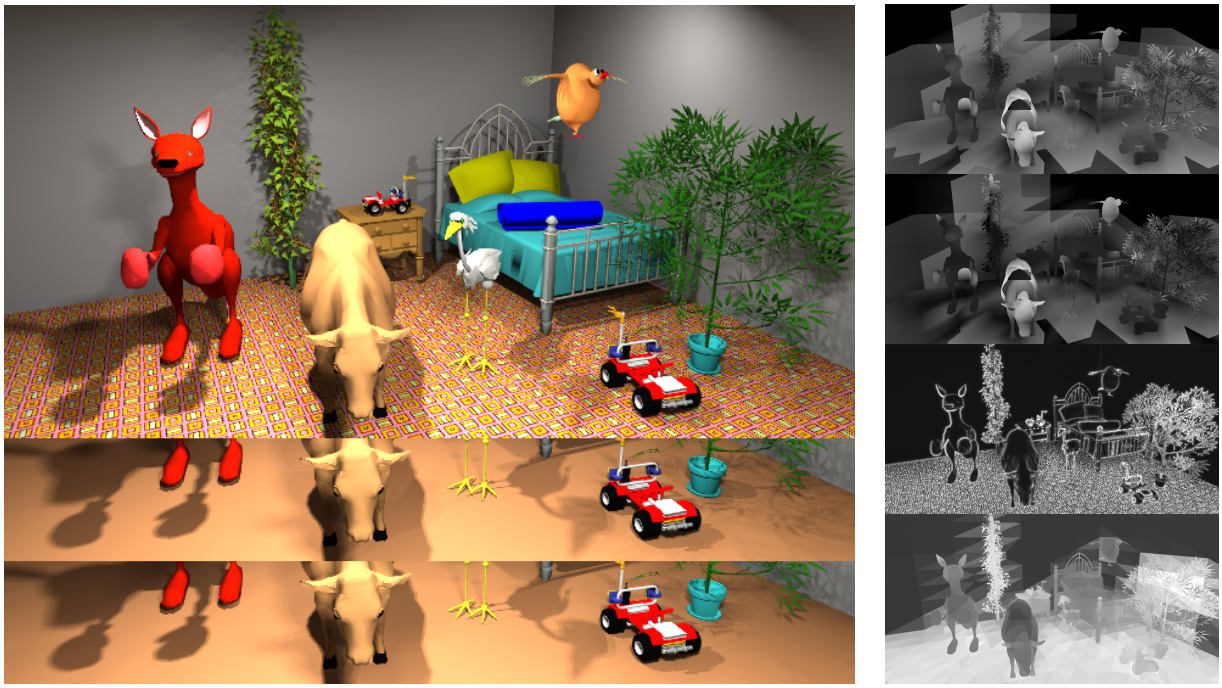

Figure 7: *Toy room, containing 250,000 triangles. The large image shows a final gather result. Below, two image details are added showing the final gather results for the same scene without textures (top) and the result from the textured scene with the texture removed (bottom) for visualizing the decreased shadow detail. On the right: number of reevaluated visibility samples without masking (top) and with masking (middle top), masking image (middle bottom) and masking per patch/grid cell (bottom).*

Figure 8: *Library, containing 500,000 triangles. Top: final gather result. Bottom row: Visibility samples without masking (left) and with masking (middle left), masking image (middle right), and masking per patch/grid cell (right).*