Thrifty Final Gather for Radiosity

Annette Scheel, Marc Stamminger, Hans-Peter Seidel

Thrifty Final Gather for Radiosity

Annette Scheel†, Marc Stamminger‡, Hans-Peter Seidel†
†Max-Planck-Institut for Computer Science www.mpi-sb.mpg.de
‡iMAGIS/GRAVIR-REVES - INRIA Sophia Antipolis www-sop.inria.fr/reves/

Abstract. Finite Element methods are well suited to the computation of the light distribution in mostly diffuse scenes, but the resulting mesh is often far from optimal to accurately represent illumination. Shadow boundaries are hard to capture in the mesh, and the illumination may contain artifacts due to light transports at different mesh hierarchy levels. To render a high quality image a costly final gather reconstruction step is usually done, which re-evaluates the illumination integral for each pixel. In this paper an algorithm is presented which significantly speeds up the final gather by exploiting spatial and directional coherence information taken from the radiosity solution. Senders are classified, so that their contribution to a pixel is either interpolated from the radiosity solution or recomputed with an appropriate number of new samples. By interpolating this sampling pattern over the radiosity mesh, continuous solutions are obtained.

1 Introduction

In the past 15 years, much research has been concentrated on improvements of the radiosity method. The computation of radiosity solutions of more than one million patches is now possible on contemporary standard PCs. However, these solutions most often do not meet high-quality demands of many commercial applications. The problem is inherent to the method: because lighting detail is generated in object space, very fine tessellation is necessary to capture fine lighting detail. Thus quality is not only a time, but also a memory issue. Furthermore, the widely used linear Gouraud interpolation is prone to Mach banding, so that a human observer perceives the tessellation structure easily. In addition, long thin triangles can lead to frayed shadow boundaries, and finally the different levels at which light is transported to a patch may lead to artifacts. To some extent, these issues can be addressed by adapting subdivision to lighting discontinuities [11, 21] or by using higher-order interpolation (e.g. [6]).

Ray-based Monte-Carlo methods usually compute the illumination at independent sample positions in image space. This point sampling allows exact lighting computations, but also makes the exploitation of coherence more difficult. As a result, illumination from big light sources or indirect light make the lighting computation expensive. For stochastic sampling as a rule 500 or 1000 samples per pixel are needed in such cases, and in some cases noise is still present.

The idea of final gathering is to combine both approaches. In a view-independent preprocess a not necessarily perfect radiosity solution is computed. Then in a second view-dependent step ray tracing is performed that recomputes parts of the illumination considered critical. One common technique is to only recompute the direct light in this pass and to add the indirect light from the finite element solution (e.g. [16, 5]). Alternatively, the radiosity solution can be interpreted as a light source definition, and the last light bounce towards the eye is recomputed with ray tracing (e.g. [2]).

1.1 Previous Work

Compared to the huge number of publications on radiosity methods, the number of publications on a high quality rendering postprocess is surprisingly low. Often, a high quality rendering step is only a final chapter for a new radiosity algorithm, e.g. [11, 17, 2, 12], but only very few papers address the issue in detail.

In [10], Kok et al. describe a method to decrease the number of shadow samples for area light sources exploiting information from a progressive radiosity pass. Their algorithm subdivides senders adaptively in order to detect completely visible or occluded parts that can be processed quickly. With a clever shadow pattern scheme, the number of samples is reduced further. To some extent, this radiosity postprocessing step is done implicitly in the radiosity computation itself by the idea of Hierarchical Radiosity [8].

Several authors proposed the use of (hierarchical) radiosity for indirect light only, and to regenerate the usually more detailed direct illumination by ray tracing. More information from the radiosity solution is exploited in the *local pass* by Linschinski et al. [11]: Form factors are recomputed for each pixel and visibility for direct illumination only. Their approach is extended by Stürzlinger [20], where the number of samples is adapted to the relative contribution of the sources and visibility is computed in a stochastic approach. Christensen et al. [2] used a final gather step for reconstructing their radiance solutions. They gather all links with a fixed number of visibility samples, but also use the cluster hierarchy. It is pointed out that the final gather requires significant computation time and that despite their conservative resampling scheme artifacts still remain visible. Smits [17] uses a fixed number of visibility tests and analytic form factor evaluations for each point. For links carrying a relative error below a certain threshold, the estimate of the link itself is taken. To avoid bias which might occur due to the threshold, Russian Roulette is used with links whose relative error is below the threshold.

The final gather step which is very briefly described for the algorithm presented in [12] extends these ideas. For each receiver, critical senders are determined, whose contribution is to be computed exactly by final gather.

Bekaert et al. [1] use the radiosity solution for defining an importance function for a Monte-Carlo ray tracing step. A similar idea later followed in [14], where a radiosity solution is used to guide a Monte-Carlo path tracer.

Finally, the photon map approach [9] is relevant, because it is essentially a final gathering approach, however it is based on a global photon tracing pass as preprocess. From the photon hits an approximate lighting solution can be computed quickly. The high quality images, however, are then obtained from a well optimized final gather step.

1.2 Idea

The idea of this paper is to better exploit the information from the global radiosity step for a more ‘thrifty’ final gather. Consider the example in Fig. 1, middle, showing a coarse radiosity solution from a candle, illuminated by a big light source on the left and a smaller one on the right. Due to the different sizes of the light sources, the shadows are very different. The left shadow has a clear outline that blurs with the distance to the object; the right shadow is blurry and completely washed out after a short distance.

The radiosity solution is obtained in a few seconds and thus of poor quality, but it is clear that for a final gather step valuable information can be obtained from it. In addition to the radiosity mesh which already captures the most important changes in illumination, the link structure contains per patch hints about shadow boundaries and the light sources responsible. Furthermore, during the radiosity computation more

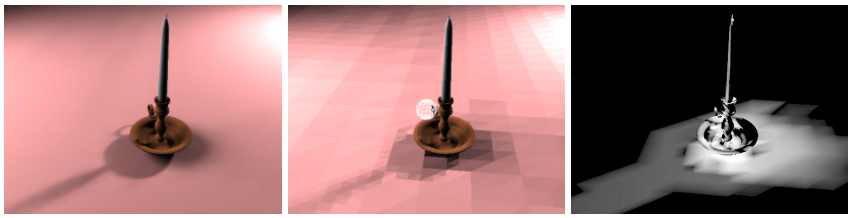


Fig. 1. Left: a candle stick casting a shadow from a large and a small area light source. Center: radiosity solution. Right: visibility sample numbers considered appropriate by our method.

approximate information has been computed with each link, such as the variation of the form factor or a shadow gradient.

In previous work, this information was exploited by recomputing partially visible links only, or by using a fixed number of samples per link, thus implicitly performing importance sampling guided by the link structure resulting from radiosity. We experienced severe problems with these simple approaches:

- At patch boundaries, the set of sampled senders and their hierarchy level change abruptly. Even for very precise solutions, these discontinuities, although small in size, remain visible.
- For indirect light a large fraction of links is partially occluded, so resampling all of them is very expensive. When not resampling indirect light, scenes with dominant indirect light essentially show the radiosity solution with all its artifacts.
- Taking a fixed number of samples results in bad importance sampling. It is true that hierarchical radiosity creates a mesh of links with similar errors, so using a fixed number of samples per link assigns roughly similar error to each sample. However, the error measured in HR for refinement decision is not necessarily the error we are interested in for final gather. Using a minimal area parameter also breaks the uniform error assumption, in particular for critical links with high variation. Links arrive at different levels of the hierarchy, making comparisons difficult. Furthermore, the error of a link must be seen in the context of all other links arriving at the same point.

Our new method tries to address these deficiencies. For a given receiver we extract a list of potential senders and estimate whether their contribution needs to be resampled or can be interpolated from the radiosity solution. If resampling is necessary, an appropriate sample number is determined. Both decisions can take into account the magnitude of the illumination, its relation to other contributing patches, visibility gradient estimates, perceptual issues etc.

By gathering this sampling information at the vertices and interpolating it over the patches again, we obtain a *continuous* solution without discontinuities at the patch boundaries. Nevertheless, the solution will not necessarily have a continuous derivative, so Mach banding can still appear. However, the goal of the method is to resample the solution so accurately, that this effect should also disappear.

Our final gather accuracy is (almost) independent of the radiosity accuracy. If the radiosity solution is already sufficient, the final gather will simply interpolate the radiosity values without further sampling. On the other hand a coarse solution will lead to a more expensive final gather step. However, the method can fail in detecting lighting detail that has not been captured at all by the input radiosity solution.

Note that the goal of our method is to enhance the final gather step as such. This

does not exclude the use of ray-tracing acceleration methods in image space, such as adaptive progressive refinement [13]. In fact, we use directional coherence maps [7] (see also results) to exploit image space coherence.

2 Algorithm

Our method is based on the results obtained by a hierarchical radiosity algorithm with clustering. In the final radiosity solution, every object is represented by a triangle mesh with an approximate radiosity value per triangle. The link mesh contains the approximate value of the form factor, its gradient, and visibility. The HR oracle is based on an estimated radiosity change over a receiver, multiplied by the receiver's area. The goal of the final gather step is to accurately recompute the last light bounce towards the camera. For now, we will handle diffuse reflection only.

2.1 Overview of the Algorithm

First, for each vertex under consideration, a list of contributing senders is created by gathering all links arriving at all levels and at all surrounding patches (Sect. 2.2 and Fig. 2, left). Then we determine for each sender, whether its contribution can be interpolated, based on the estimated magnitude of the change in illumination it will introduce (Sect. 2.3). Furthermore, we distinguish between variations caused by the form factor and the visibility term. For a low variation the corresponding term will be interpolated, otherwise an appropriate sample number for final gathering is computed and stored with the sender (Sect. 2.4 and Fig. 2, middle).

In order to perform final gathering at a visible object point, the surrounding vertices are considered. The contributions of all senders that are marked as 'smooth' in all vertices are interpolated from the vertices to the point with its barycentric coordinates. For all other senders, samples are spawned, where the number of samples is again interpolated from the vertices (Fig. 2, right). Sect. 2.5 describes our interpolation process that leads to a continuous final gather result.

In the following, we describe the single phases of the method in more detail, discuss alternatives and justify our choices.

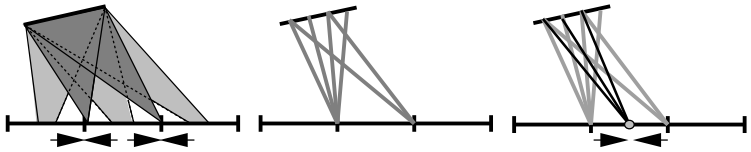


Fig. 2. Overview. Left: Collecting links from neighboring patches. Middle: Determination of sample numbers for each vertex. Right: Computation of illumination for a point inside a triangle.

2.2 Collection of Links for Vertices

In the radiosity solution links might arrive at different levels in the hierarchy at the receiver. Therefore, prior to the main algorithm, the links are 'pushed down' to leaf level on the receiver. During this push we recompute the form factor for the new links, but reuse the much more expensive visibility term. This recomputation avoids artifacts from the constant push step that typically appear when light is gathered at different hierarchy levels.

The algorithm begins by collecting the links from all patches surrounding a vertex, such that the vertex obtains a list of all senders. The information of the links is either

averaged (for example for the irradiance) or the maximum is taken (for example for the visibility and form factor gradients). In this step links with visibility zero are neglected, which can be inaccurate if visibility is sampled by casting rays. However, a 'false reject' of a sender partially visible to the vertex requires that for all surrounding patches the sender was wrongly classified as completely occluded.

2.3 Classification of Senders

Usually, illumination exhibits strong coherence; hard-to-capture variations in form factor or visibility are only produced by a small portion of senders. These do not necessarily have to be light sources but can also be strongly illuminated patches creating strong indirect illumination. In either case, our algorithm tries to identify those "trouble makers" and resample them.

From the existing radiosity solution and from values computed for vertices information can be extracted to drive an **oracle** to detect senders with difficult contributions. Our oracle estimates the magnitude of the change in radiosity around a vertex due to form factor and visibility variations[1]. The threshold, which decides if this magnitude is small enough for interpolation, should respect human perception, because differences in color are not perceived equally well for all levels of illumination. In our implementation we use the CIELAB space [3] [2], because it is designed to be perceptually uniform and a color difference measure is defined which outputs JNDs (Just Noticeable Differences). Given a user specified JND, we take the radiosity values from the radiosity solution as a basis to determine which differences in illumination are acceptable.

Our first attempt was to decide according to the gradient of the illumination at the vertex. Although high gradients are a valuable hint for non-smooth illumination, the absolute difference turned out to be more significant. Additionally, it is much harder to find an intuitive threshold for gradients.

Variations in radiosity can be caused either by the form factor or the visibility term, in the diffuse case. For example, a light source could cause strong variations of the form factor, but if no occluders are present, only the form factor has to be recomputed, while visibility can be interpolated. Therefore, we estimate form factor and visibility variation separately and decide individually if form factor or visibility can be interpolated. However, the two functions depend on each other: if a patch is illuminated only weakly, shadows will be less obvious. On the other hand, a strong variation of illumination may be canceled out by a shadow. Therefore, if the variation of one term is measured the other term is kept constant.

We have developed two different oracles. The first, which compares illumination values of neighboring vertices, is more time consuming but turned out to be more reliable than the faster second, which re-uses link information from the radiosity solution.

Estimating Variation from Vertex Differences. The radiosity at a vertex c can be computed precisely using the sample distribution which will be described in Section 2.4. The radiosity of each sender is compared with the corresponding values of the neighboring vertices v_i to find out how fast the illumination changes within the neighborhood of vertex c.

[1] The term *variation* will be used in the following not in its mathematical meaning, but to describe *magnitude of change in radiosity.*

[2] Prior to the conversion to CIELAB a tone mapping should be applied to the color, which is not done in this implementation yet.

We determine the variation of the visibility term $D_s^{vis}(c)$ by computing the maximum difference between the radiosity at the center vertex due to sender s and the radiosity at all neighbor vertices v_i due to the same sender. As explained before, the form factor is kept constant for this comparison. $D_s^{vis}(c)$ is then

$$D_s^{vis}(c) = B_s F_s(c) \max_i (vis_s(c) - vis_s(v_i))\rho, \tag{1}$$

where B_s is the sender's radiosity, F_s the form factor between c and s, $vis_s(c)$ is the percentage of s which is visible from p, and ρ is the reflectivity of the receiving patch. The variation of the form factor $D_s^F(c)$ is computed analogously:

$$D_s^F(c) = B_s \max_i (F_s(c) - F_s(v_i)) vis_s(c)\rho. \tag{2}$$

For both, $D_s^{vis}(c)$ and $D_s^F(c)$, a perceptual error threshold can be used. The measure can only determine if there are strong changes expected around the vertex or not, but not if the shape of the function is suited for interpolation. Consequently the measure is rather strict: it will select those contributions which can safely be interpolated, but does not detect all contributions which would also be suitable for interpolation. Of course, it is possible that features inside a triangle are missed: we can only capture what is possible with the given radiosity mesh.

Estimating Variation from Links. The links of the radiosity solution give us a cheaper estimate for the change in illumination across a receiving patch. Again, the maximum of the variation of all patches surrounding a vertex is used as estimation of the variation at a vertex.

For the hierarchical refinement an estimation of the variation of the form factor is needed anyway. For example, we use Bounded Radiosity [19], which gives a conservative lower and upper bound for the form factor. Thus, the change in radiosity due to the form factor $D_s^F(p)$ over patch p due to sender s is:

$$D_s^F(p) = B_s(F_s^{\text{upper}}(p) - F_s^{\text{lower}}(p)) vis_s(p)\rho. \tag{3}$$

The vertex variation can then be computed from the maximum variation of the surrounding patches, e.g. $D_s^F(c) = \max_p(D_s^F(p))$.

If partial visibility between a sender and a receiver is detected in a link, the maximum possible change in radiosity goes from zero up to the unoccluded illumination. This gives a conservative bound on the variation due to visibility.

Additionally, we estimate the size h of the penumbra by taking into account geometric considerations (see next paragraph). With this penumbra size the bound can be tightened further: if h is larger than the patch the full cut-off will not take place inside the patch. If the visibility produces a function with gradient $B_s F_s(p)\rho/h$ then the change inside the patch with size s_r is only $B_s F_s(p)\rho s_r/h$. In summary the visibility variation $D_s^{vis}(p)$ is determined by:

$$D_s^{vis}(p) = \begin{cases} B_s F_s(p)\rho s_r/h, & h > s_r \\ B_s F_s(p)\rho, & \text{else} \end{cases} \tag{4}$$

As before, the vertex visibility variation $D_s^{vis}(x)$ is then the maximum variation of the surrounding patches. Using the maximum of all surrounding patches decreases the risk of missing an occluder.

Estimating the Size of the Penumbra. The size of the penumbra which might be produced by a sender is estimated using simplified geometric configurations. This was done before in a similar way by [18].

The size h of the penumbra can be approximated by (see Fig. 3):

$$h = d_s \frac{\cos(\theta_s)}{\cos(\theta_r)} \frac{||s-b||}{||b-r||}, \tag{5}$$

where d_s is the maximum diameter of the sender, $\cos(\theta_s)$ and $\cos(\theta_r)$ are the angles between the direction between the sender's and the receiver's centers and the corresponding surface normals.

During the visibility computation of the radiosity algorithm we store the occluder with hit point closest to the receiver. For this closest hit point we compute the relation between distance of the sender to the blocker $||s-b||$ and of the blocker to the receiver $||b-r||$.

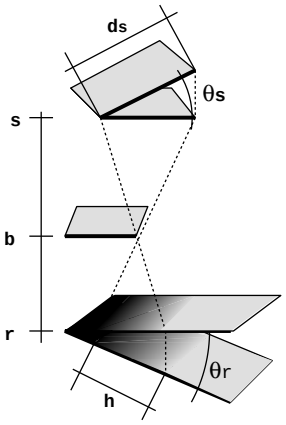


Fig. 3. Size of the penumbra.

Addition of Sender Variations. So far, only single sender contributions have been investigated. But the contributions of different senders may influence each other and in the worst case several small variations may sum up and thus cause a significant variation.

To capture this worst case we select the candidates for interpolation in the following way: The variations of all senders of a vertex are sorted in decreasing order. Then, from the total sum of all variations, we subtract the single variations until the sum drops below the error threshold. The remaining contributions can then safely be interpolated, because even if they should all sum up, they will be below the error threshold. Since the highest variations are subtracted first, we ensure that the most important ones are not missed. On the other hand, this method may be too strict, because in many cases the contributions do not sum up and might even cancel out.

2.4 Number of Samples at a Vertex

If the classification procedure determines that it is better to re-compute a contribution exactly, the next step is to find the number of samples which should be used for each sender. This is not exactly true for classification by vertex comparison: the number of samples is determined first to compute the illumination at a vertex exactly in this case.

To determine the irradiance $I(x)$ at a point x the integral over all sending surfaces has to be evaluated:

$$I(x) = \sum_{\text{Senders } s} \int_s B(y) F(x,y) vis(x,y) dy, \tag{6}$$

where $B(y)$ is the radiosity of the sender, $F(x,y)$ the point-to-point form factor, and $vis(x,y)$ is true if point y is visible from point x.

We want to evaluate this integral for each vertex within a user given precision. As B we take the radiosity solution, this means the final gather recomputes the final reflection. The form factor and the visibility term are re-evaluated by randomly choosing N_s sample points on each sender.

In our algorithm, N_s is chosen in accordance to the magnitude of the irradiance at x due to this sender. From the radiosity solution we already have an estimate for this

value—not for each vertex but for each patch. The relative contribution of a sender s to a receiving patch p compared to the total irradiance at this receiver is $I_s(p)/\sum_S I_s(p)$, where the irradiance $I_s(p)$ can be taken from the links, if the form factor is stored. We use the unoccluded irradiance, because the visibility term would reduce the number of samples in the presence of occluders, although this situation needs a high number of samples to capture the shadow boundary.

Given a total sample number N to approximate integral (6) the number of samples $N_s(p)$ for sender s and patch p is given by $N_s(p) = NI_s(p)/\sum_S I_s(p)$. To finally determine the number of samples N_s for a vertex x the average of the number of samples proposed by each neighboring patch sharing the vertex is taken. Only if all senders are classified as "to be re-sampled" the full sample number N will be used. In general, our algorithm manages with only 30% of N (see Sect. 3).

One could argue that it would be more efficient to use a total number of samples N which is not fixed but proportional to the aggregate irradiance, this generally means fewer samples for darker areas. In our case, we base our decision on interpolating contributions on the values computed for the vertices (see Sect. 2.3); therefore having a fixed precision for the vertices is more advantageous. Furthermore, it may be inaccurate to lower the precision for dark areas, because differences in illumination are perceived more easily there.

In many cases the sending patch will be a triangle or another primitive for which an analytic patch-to-point-form factor is known. Although this formula is also expensive to compute—it contains three $\arccos$ evaluations for a triangle for example—it might be faster than sampling if a high number of samples is used. However, our measurements (see Sect. 3) show that the time spent for form factor computation is low compared to the time spent for visibility computation.

2.5 Rendering a Triangle

After the preprocess each vertex has a list of senders with either precomputed illumination for interpolation or a set of sample positions for exact computation.

From the number of samples for the vertices the sample numbers for points inside a triangle can be determined. This is again done per sender: For a point x the number of samples for a sender s is the sum of the three vertex sample numbers for s weighted by the barycentric coordinates of point x.

Note, that we use real numbers as sample numbers to achieve smooth transitions. Having $f \in R$ samples, actually $n = \lceil f \rceil \in N$ samples will be evaluated. Each sample is weighted by one, except for the last which gets weight $f+1-n$. Although this results in overhead, it ensures that the sample number is continuous inside the triangle enabling a smooth representation of illumination. Furthermore, this methods works well only if the sample positions are fixed for a sender, i.e., if the i-th sample is always located at position y_i on the sender.

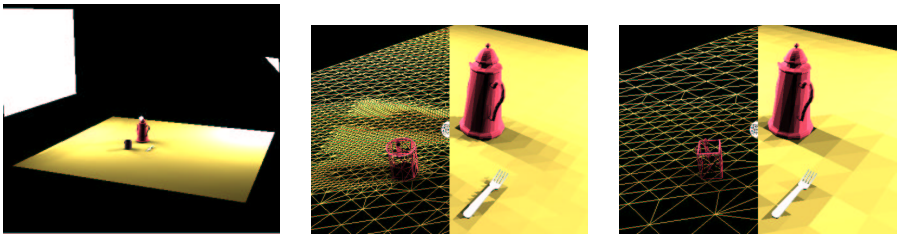
It is possible that a sender is classified differently by two neighboring vertices. In this case, we compute the result according to the first as well as according to the second classification and interpolate the two results, weighted by the barycentric coordinates, to obtain a continuous representation.

3 Results

3.1 Test Series

In order to test the behavior of our method, we examined the influence of different parameters on execution time and visual quality of the result. Our test scene is shown in Fig. 4(a). Three objects on a table are lit by two area light sources, a very large one and a smaller one. Both create large variations in irradiance with respect to form factor and visibility. We used the large light source as a representative for indirect light, but by making it a light source we can generate a reference solution by ray tracing.

The large penumbrae result in big problems for standard ray tracers and on final gather approaches. Obtaining an image without visible noise requires about 1000 light source samples per pixel; our reference solution was computed with Monte-Carlo ray tracing with 2000 samples in 90 minutes.[3]



(a) Test scene overview (b) Radiosity mesh 1 (c) Radiosity mesh 2

Fig. 4. Test scene and radiosity solutions used for final gather

Fig. 5 shows three final gather results with sample number 800. The first one was obtained from the radiosity solution in Fig. 4(b) with small final gather thresholds in 578s. The result (top row) is essentially indistinguishable from the reference solution. Only the difference image (bottom row, scaled by 10) reveals errors in particular in the penumbrae. On average, visibility was resampled by 118 samples per pixel, illumination by 65 (out of 800 possible). 81% of the time were spent on the exact sampling, the remaining 19% are used for sample classification etc. (algorithm overhead).

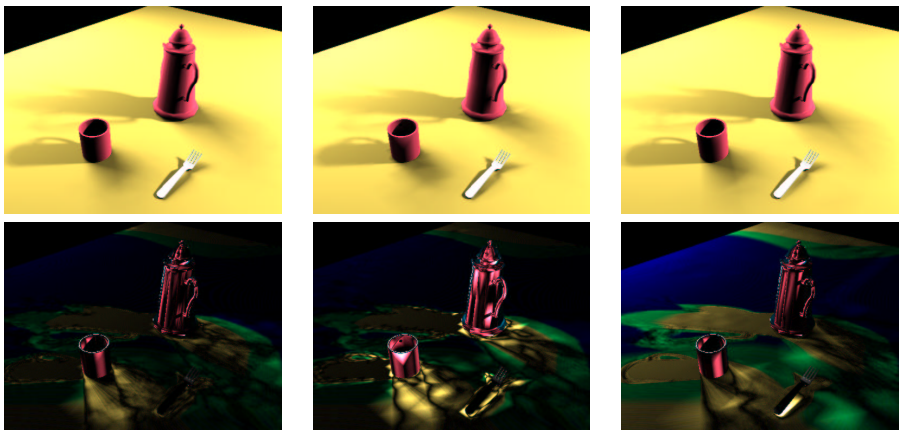
The second column shows a less accurate solution with only 56 exact samples per pixel for visibility and 23 for the form factor on average. Artifacts become visible, but the time also decreased to 289s with an overhead of 20%. Sample numbers were determined using link information instead of the more accurate criterion (see Sect. 2.3).

For the right column, the same parameters as for the first image were used, but a coarser radiosity solution (see Fig. 4(c)) was taken as input. The resulting image has almost the same quality as the first one, but rendering time went up to 985s. The number of exact samples for visibility and form factor went up to 265 and 234, respectively. The overhead went down to 7%.

3.2 Performance for More Complex Scenes

To test the performance and behavior of the final gather for more complex environments we used three scenes with different lighting characteristics. In the first scene “office” (taken from the Radiance homepage) indirect illumination is present but not very obvious (color bleeding on the wall from the table, see color page, first row). In contrast, the

[3] All tests in this section were perfomed on a Linux Pentium PC with 733MHz, those in the next sections on a 1.1GHz Pentium



578s 289s 985s

Fig. 5. Final gather results. Left: fine radiosity solution with accurate final gather. Middle: fine radiosity solution with less accurate final gather. Right: coarse radiosity solution with accurate final gather. Lower row: difference to reference solution.

second scene "work and rest" (w.a.r.) is mostly lit indirectly including shadows from indirect light (see color page, second row). Note that, although we selected 400 (importance weighted) samples and the algorithm recomputed the highest possible sample number in these regions, still small artifacts in the indirect shadows of the plants are visible. The last scene, "public library" has raised complexity (see color page, last row). It contains 83 mostly big and lengthy area light sources, casting all kinds of sharp and blurry shadows on the floor, which causes significant problems on any Monte-Carlo and final gather approach. The HR solutions contained 7,800, 53,252, and 97,986 triangles, for office, w.a.r, and publib, respectively.

Table 1 shows the sampling statistics. The average proportion of recomputed samples for form factor and visibility varies from 25% for the office to 36% for the public library. The (v) and (l) after the scene denote whether the vertex comparison or the link information was used to find senders suitable for interpolation, respectively.

scene	total	avg. vis	avg. ff
office(v)	400	101	148
office(l)	400	130	230
w.a.r.(l)	400	126	122
publib(l)	1000	361	230

Table 1. Sample numbers for test scenes. Total: max. number of importance weighted samples per pixel, avg. vis: recomputed visibility samples, avg. ff: recomputed form factor samples.

Table 2 lists the timings for the three scenes, split up into the single computation steps. The time to classify for each sender if its contribution should be interpolated or re-sampled, depends strongly on the chosen method. As expected, reusing the link information is much faster than comparing the vertices. This is mainly because for the vertex comparison the corresponding senders for all neighbor vertices have to be found. Both methods turned out to be reliable in finding shadows and other lighting details, even for quite coarse meshes. In some cases the link information was slightly less reliable. The reliability of the link information depends also on the number of test rays

used for HR. Furthermore, the bounds computed from the links on the form factor and visibility are not very tight. Therefore generally more samples are used (see Table 1), which might cancel out the time savings during the classification process.

The table column 'vertex' contains all other preprocessing steps: collection of links, determination of sample numbers, and computation of vertex irradiance. The timings here show that for highly refined radiosity solutions the vertex preprocess time slows down the algorithm.

Finally, the time for rendering the interior points was split into time for visibility and form factor computation and the remaining time which is spent for collecting the senders and interpolation. In general, by far most time is spent for visibility computation (75–80%), only a very small amount for form factors (1–10%) and the overhead is approximately 15–20%.

scene	HR	push link	classific.	vertex	interior			total
					vis	ff	rest	
office (v)	25	10	54	45	160	28	48	345
office (l)	25	10	0.1	40	217	38	52	357
w.a.r. (l)	28	3	0.1	140	518	34	51	745
publib (l)	185	37	1	405	582	15	182	1222

Table 2. Timings in seconds for test scenes (see color plates)

The images shown on the color plate were created in combination with the Directional Coherence Maps (DCM) [7]. In our case the DCMs were used to obtain antialiased images (36 fold oversampling) with only a moderate increase in computation time. The quality of the lighting reconstruction was not affected. The timings in Table 2 are without DCM, with DCM the whole computation took 508s, 1489s, and 4435s for office, w.a.r, and publib, respectively.

4 Conclusion and Future Work

We presented an improved, adaptive final gather step. For all pairs of receivers and senders an oracle decides whether the sender's contribution can be interpolated or needs resampling. This classification is done separately for the form factor and visibility term. The required information can be taken from the radiosity solution or it can be obtained more precisely from a full final gather at the vertices. If resampling is necessary, the number of samples is chosen adaptively. By interpolating the sample pattern, continuous solutions are obtained.

The accuracy of the final gather step is chosen independently of the radiosity solution. More sophisticated error measures can be used than in standard radiosity, for example perception based criteria, that require a more 'global view' than that of a radiosity refiner. Integration of perception is planned to be enhanced in the future, for example reducing accuracy where textures can mask out artifacts similar to [15] or for patches creating frequencies in a range where sensitivity for contrast goes down.

The performance of the final gather scales with the accuracy of its input: for accurate radiosity solutions it is usually faster than for a coarse radiosity solution. This is true as long as single patches of the radiosity solution are larger than a few image pixels. If this is not the case, the overhead of the sample classification becomes dominant. Here, a faster classification procedure which compares total illumination and not the illumination of each sender would be advantageous. For objects containing triangles in sub-pixel space it is probably necessary to switch from the polygonal mesh to a differ-

ent representation (e.g. use face clusters [22] or a 3D grid bounding complex objects [4].) For very complex scenes the push step for links also might become a bottleneck both in terms of time and memory. Therefore, we would like to investigate if the final gather can work directly on the hierarchy of objects and links as created by the HR and not only at leaf level as was done in [12].

5 Acknowledgments

The first author is supported by the ESPRIT Open LTR Project 35772 "SIMULGEN", Simulation of Light for General Environments. We would also like to thank George Drettakis for fruitful discussions and proof-reading and the reviewers for their valuable comments. iMAGIS is a joint project of CNRS/INRIA/UJF/INPG.

References

[1] 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] 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.

[3] CIE. Recommendations on uniform color spaces–color difference equations–psychometric color terms. Technical Report 15.2, CIE, 1996.

[4] 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.

[5] Reynald Dumont, Kadi Bouatouch, and Phillipe Gosselin. A progressive algorithm for three point transport. *Computer Graphics Forum*, 18(1):41–56, 1999.

[6] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 221–230, 1993.

[7] Baining Guo. Progressive radiance evaluation using directional coherence maps. In *Computer Graphics (SIGGRAPH '98 Proceedings)*, pages 255–266, 1998.

[8] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 197–206, 1991.

[9] 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.

[10] 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.

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

[12] 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.

[13] James Painter and Kenneth Sloan. Antialiased ray tracing by adaptive progressive refinement. In *Computer Graphics (Proc. SIGGRAPH '89)*, volume 23, pages 281–288, 1989.

[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.

[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.

[16] P. Shirley. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois, 1991.

[17] Brian Smits. *Efficient Hierarchical Radiosity in Complex Environments*. Ph.d thesis, Cornell University, 1994.

[18] Cyril Soler and François X. Sillion. Fast calculation of soft shadow textures using convolution. In *SIGGRAPH 98 Conference Proceedings*, pages 321–332, 1998.

[19] Marc Stamminger, Philipp Slusallek, and Hans-Peter Seidel. Bounded radiosity - illumination on general surfaces and clusters. *Computer Graphics Forum (Eurographics '97)*, 16(3):309–318, 1997.

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

[21] V. Volevich, K. Myszkowski, A. Khodulev, and E. Kopylov. Using the visible difference predictor to improve performance of progressive global illumination computation. *ACM Transactions on Graphics*, 19(1):122–161, April 2000.

[22] 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.

Fig. 6. Results. Upper small image: number of samples (red: visib., green: formfactor). Lower small image: radiosity solution. Images sizes (top to bottom): 424x363, 512x384, 427x363.