Eurographics Symposium on Rendering 2003, pp. 1–9
Per Christensen and Daniel Cohen-Or (Editors)

Interactive Global Illumination in Complex and Highly Occluded Environments

Ingo Wald Carsten Benthin Philipp Slusallek

Computer Graphics Group, Saarland University
{wald,benthin,slusallek}@cs.uni-sb.de

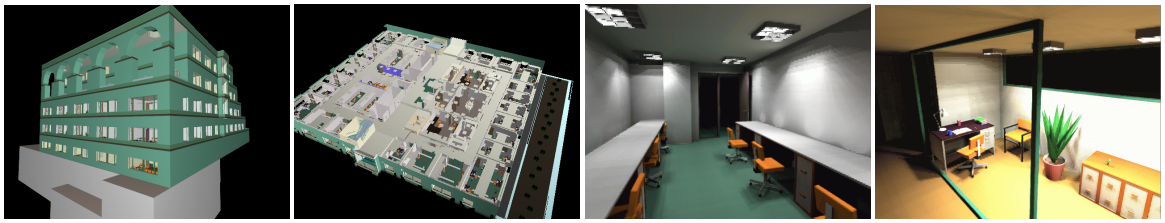

Figure 1: *"Soda Hall", a 7 storey fully furnished building containing 2.5 million triangles and 23,256 light sources. a.) The entire building, simply shaded, b.) Overview of a single floor c.) and d.) Inside, with full global illumination. With our proposed technique, the right images run at 2-3 frames per second on 22 CPUs allowing to interactively explore the entire building with full global illumination.*

Abstract

Global illumination algorithms have traditionally been very time consuming and were only suitable for off-line computations. Recent research in realtime ray tracing has improved global illumination performance to allow for illumination updates at interactive rates. However, both the traditional off-line and the new interactive systems show significant limitations when dealing with realistically complex scenes containing millions of surfaces, thousands of light sources, and a high degree of occlusion.

In this paper, we present an importance sampling technique that has specifically been designed for such environments. Our method maintains a rough estimate of the importance of each light source with respect to the current view using a crude path tracing step. This estimate is then used to focus computations to the most important light sources. In addition to speeding up the computation our approach minimizes the working set of the ray tracer by only touching geometry that is relevant to the current view. This allows us to directly and efficiently render scenes such as entire buildings with many thousands of light sources at interactive rates with full global illumination.

1. Introduction

Global illumination – the task of simulating the transport of light in a scene – is an important task for many practical applications, e.g. in product design, virtual prototyping, and architecture. Due to the extensive research over the last two decades most lighting effects can now be simulated with sufficient accuracy – diffuse, direct, and indirect lighting, shadows, reflections and refraction, high-quality caustics, and even volumetric and more advanced lighting effects. However, these computations require extensive computations and are usually far from interactive.

With recent advances in realtime ray tracing and appropriately designed global illumination algorithms, it has recently become possible to render complex scenes with the most important global illumination effects at interactive rates [18, 2]. However, the disciplines mentioned above often require the realistic visualization of entire planes, ships, complete buildings, or construction sites. Such scenes often consist of many individual rooms and contain millions of triangles and hundreds to thousands of light sources (see Figure 1), that cannot easily be handled by todays algorithms.

Due to a high degree of occlusion, most of the different rooms in such scenes are typically influenced by only few of the a small fraction of all light sources. For example, a room on the 7th floor of the building in Figure 1 will hardly be illuminated by a light bulb in the basement. Large numbers of light sources are a challenge for most off-line and interactive algorithms because many samples are computed without a significant contribution to the final image.

While it has been common to optimize the rendering process by manually disabling irrelevant light sources for off-line processing, this is not an option for interactive applications where the set of relevant light sources can changes from frame to frame.

In this paper, we present an offline and an online method that exploits the special characteristics of such scenes by automatically determining the relevant light sources. We estimate the visual importance of each light source for the final image and use this estimate for efficiently sampling only the relevant and visible light sources. While we do not want to compromise on quality for the offline rendering setting, we tolerate a reasonable amount of artifacts in the interactive setting as long as the general impression remains correct.

We start with a detailed analysis of the problems that arise when calculating global illumination in complex and highly occluded scenes in Section 2, together with a short discussion of previous work in Section 3. In Section 4 we present the basic idea of our method and discuss its use in Section 5. Finally, we conclude and discuss future work in Section 6.

2. Global Illumination in Realistically Complex Scenes

Realistic scenes such as those listed above share a number of characteristics: Massive geometric complexity of up to millions of triangles, hundreds to thousands of light sources, and high occlusion between differnet parts of the scene.

2.1. Geometric Complexity

Realistically complex scenes often consist of millions of triangles in order to accurately model the geometric detail (e.g. detailed pencils in the "Soda Hall" scene). Such geometric complexity has always been problematic for radiosity-style algorithms that have to store illumination information with the geometry of the scene. While clustering [15] and the use of view importance [1] do help in such scenes, these algorithms must still sample and process the entire scene geometry.

For all ray-based rendering algorithms – e.g. (bidirectional) path tracing [6, 16, 10], instant radiosity [18, 7], or photon mapping [5] – the pure number of triangles is theoretically less of an issue, as such algorithms are sublinear in the number of triangles [17, 12]. This allows to efficiently render even scenes with millions of triangles [19, 13] (see Figure 2) and at least in theory makes such algorithms mostly independent of the scene complexity. In practice, however, this holds true only if a high coherence of the rays can be maintained [17].

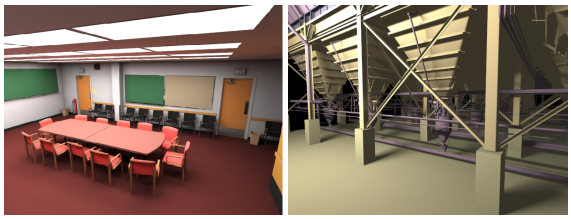

Figure 2: *Neither geometric complexity, nor a moderate number of light sources is a problem for e.g. the instant global illumination method. Left: The "conference" model with 202 light sources, interactively illuminated at 5-10 fps. Right: instant global illumination in the 12.5 million triangle "Powerplant" at 2-3 fps. Note that the power plant contains only a single, manually placed light source, and thus shows little occlusion and high coherence.*

However, as soon as the rays start to randomly sample the entire scene – as done by virtually all of todays global illumination algorithms – even moderately complex scenes won't fit into processor caches, resulting in a dramatic drop of performance. *Really* complex scenes may not even fit into main memory, leading to disk thrashing if sampled incoherently.

2.2. Many Light Sources

The second important cost factor is the large number of light sources. In reality, any room in a building usually contains several different light sources, resulting in hundreds to thousands of light sources. Many algorithms require to consider all light sources (e.g. by starting paths or particles from each of them), thus requiring a prohibitively large number of rays to be shot for a single image. If these lights all contribute roughly equally to each point, even severe subsampling works nicely (see Figure 2). However, it no longer works in our setting of highly occluded models.

Even worse, however, is the fact that the light sources are usually scattered all over the model such that it becomes hard to avoid sampling the entire geometry. This is especially true for those kinds of algorithms that have to start rays, paths, or particles *from the light sources*. Unfortunately, this applies to almost all of today's global illumination algorithms.

2.3. High Occlusion

Finally, the above mentioned scenes are usually highly occluded, and only few light sources will actually contribute to each given point. For many algorithms, this results in wasted computations. For example, algorithms that work by tracing particles from the light sources will waste most of their time tracing particles into distant parts of the scene where they will not at all contribute to the current image.

Similarly, all algorithms that require to find valid connections between a surface point and a light (e.g. path tracing, bidirectional path tracing, or any algorithm computing direct illumination separately) have to generate lots of costly samples just in order to find the few unoccluded connection. Generating enough unoccluded paths to light sources to achieve a sufficient image quality requires to shoot far too many rays for reasonable rendering performance.

2.4. Conclusions

While each of the problematic characteristics – high geometric complexity, large number of light sources, and high occlusion – can be handled relatively well by at least one of the available techniques, their combination is hard to handle

for any of these algorithms. In summary we need an algorithm that is based on ray tracing to efficiently handle the geometric complexity. Furthermore it must generate highly coherent rays that touch only the relevant parts of the model. Finally, it *must not* start rays from light sources or at least minimize these rays in order to reduce the working set of the algorithms for very complex scenes.

3. Previous Work

Each of the individual problems has received significant previous research, so that we can only review the most important contributions. For handling complex models all different kinds of ray tracing have proven to be very effective [12, 17, 19], as long as coherence of the rays is high and the working set of the rendering algorithm remains relatively small. Pharr [13] demonstrated an out-of core system that can handle millions of triangles even for simulating global illumination. However, this system is not easily applicable to an interactive setting and did not specifically target highly occluded scenes.

For radiosity-style algorithms, hierarchical radiosity [3] and clustering [15] have been introduced to improve the performance of radiosity methods for complex scenes. View importance [1] has been used to concentrate computations to parts of the scene relevant to the image. However, this approach still iterates through the entire model in order to check for the propagation of importance. All these algorithms are at least linear in the scene complexity and it has been difficult to adapt these algorithms for interactive use.

For ray-based systems, Shirley et al. [14] have proposed several importance sampling technique for the efficient handling of many luminaires, that are a basic requirement for any high-quality global illumination algorithm. However, these techniques do not account for visibility and thus are not well suited for highly occluded scenes where the importance of each light source is much more determined by its visibility than by its extent and orientation.

Ward et al.[20] introduced an algorithm to select the most relevant light sources during rendering of a single frame while the contribution of other light sources was estimated without additional visibility tests. Our approach uses the same idea but extends it to deal with complex scenes with high occlusion in an interactive context.

To account for occlusion, both Jensen et al and Keller et al [4, 9] have proposed to use a preprocessing step for approximating the direct illumination using a kind of photon map. During rendering, this information could be used to estimate the importance of a light source. This allows for efficient importance sampling by concentrating samples to the actually contributing light sources. However, their methods require to store a photon map, and is not easily applicable to an interactive setting. Furthermore, the preprocessing step required to first emit photons from *all* scene lights, which is not affordable for highly complex environments.

4. Efficient Importance Sampling in Complex and Highly Occluded Environments

Our approach is mainly targeted for realistically complex that combine high geometric complexity, many lights, and high occlusion. Several examples of such scenes – the same scenes we will use in our experiments lateron – can be seen in Figure 6: Both “ERW10” and “Ellipse” have only moderate complexity, but already contain many lights and high occlusion. Additionally, “Soda Hall” is a more realistical model of an existing building at Berkeley University, featuring 2.5 million triangles in highly detailed geometry, and 23,256 light sources scattered over seven stories of dozen different rooms each.

As discussed above, high geometric complexity can be handled well by ray tracing systems, as long as the costly processing of mostly occluded lights, and random sampling of the whole model is avoided. Achieving high performance in such scenes requires us to efficiently determine the non-contributing lights without sampling the entire scene.

As a solution, we have chosen to use a two-pass approach: In a first step, we use a crude estimation step to roughly determine the importance of the different light sources for the image, and thus to identify the important light sources.

In the second step, this information is used for improving the rendering quality with importance sampling of the light sources. This not only concentrates samples on most likely unoccluded lights, but also avoids sampling parts of the scene that are occluded.

Step 1: Path Tracing to Estimate Light Contributions

For the estimation step, we have chosen to use an eye path tracer with relatively low sampling rate. Though a path tracer may at first seem unsuited for this task (being well known for its noisy images), there are several good reasons for our choice: First, a path tracer is trivially parallelizable, which is an important feature to be applied in an interactive setting. It also easily allows to trade quality for speed by just changing the number of paths used per pixel.

Second, a path tracer only builds on ray tracing, and does not need any additional data structures. Thus, geometric complexity is not a problem as long as the rays remain mostly coherent. This coherence is, however, not a problem either. While path tracers are known for their *lack* of coherence (because the randomly chosen paths sample space incoherently), this is no longer true on a coarser scale: As a path tracer is purely view-importance driven, it will only sample geometry that will likely contribute to the image.

Of course, the path tracer eventually has to sample all lights with shadow rays, as it can not know which lights are unimportant. However, these shadow rays, if shot *from* the path *towards* the light source, will either reach the light source (in which case the light source is important) or be

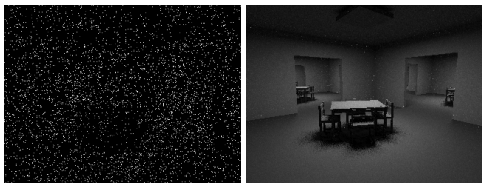

Figure 3: *Quality of the estimate in the ERW10 scene. Left: Estimate as results from a path tracer using a single sample per pixel. Right: The same image rendered with 1024 samples per pixel. Though the estimate image is hardly recognizable, the contributions of the light sources – over the whole image – are estimated correctly up to a few percent.*

blocked in the visually important part of the scene in which it was started. Thus, the actual footprint of data touched by the path tracer directly corresponds to those parts of the model that are actually important.

One potential problem with using a path tracer is that pure path tracing usually results in very noisy results, and requires lots of samples for reliable results. Though this is undoubtedly true for the whole image, we are not interested in the actual pixel values, but only in the *absolute* importance of light sources for the entire image. Then, even when shooting only a single path per pixel, rather high sample rates and reliable results can be produced.

For example, if we render the ERW10 scene which contains 100 light sources at a resolution of 640×480 with only one sample per pixel, then more than 3,000 samples will be used per light source. Thus, even if a path-traced image with as few samples may be hardly recognizable at all, reasonably reliable estimates can still be expected (see Figure 3).

Of course, some noise remains in the form of variance in the estimate. This, however is not a problem as the estimated importance will never be visualized directly but will only be used for importance sampling. A strongly varying estimate may, however, become problematic if used in an interactive context with low sampling rates for light sources, where it may lead to temporal artifacts like flickering. This problem will be addressed in more detail in Section 5.

Step 2: Constructing an Importance Sampling PDF

The method used for constructing the PDF from the path tracing information can greatly influence the rendering process. To obtain an unbiased estimate the PDF used for importance sampling should never be zero for any actually contributing light. Though this could be easily guaranteed by just assigning a certain minimum probability for each light source, this would result in many light sources being sampled and their surrounding geometry being touched. At the expense of being slightly biased, this can be avoided by thresholding the importance of light sources. This will effectively 'turn off' light sources with a very small contribution. Even though being biased, this thresholding is in practice hardly noticeable if the threshold is chosen small enough.

Step 3: Importance Sampling during Rendering

After computing the PDF we need to use it during rendering. For most algorithms, the only modification to the rendering phase is to simply replace the existing PDF used for sampling the light source.

While we want to eventually use our technique for interactive applications, it also works in an offline context: As a proof of concept we have first applied it to a simple bidirectional path tracer[16, 10]. Integrating our method was trivial, as only the PDF for sampling the start point for the light ray had to be modified. For the estimation step, we usually use only a single path per pixel. As this is rather small compared to the 16 – 64 bidirectional paths during rendering, the cost of the estimate does not play a major role for the offline rendering application. Of course, all comparisons between new and old version do include the estimation overhead.

Using our method allows to efficiently concentrate samples to important light sources. For ERW10, this translates to using only 8 instead of 100 lights. For Soda Hall the benefit is even bigger, reducing the number of lights in some views from 23,256 to only 66. Using the same number of paths, this translates to visibly better quality, as can be seen in Figure 4. This can also be measured in terms of RMS error to a master image, where – after the same rendering time – the new method produces significantly less error (see Figure 5). While the renderer used in these experiments is rather simplistic, we believe that these results should similarly translate to more sophisticated renderers.

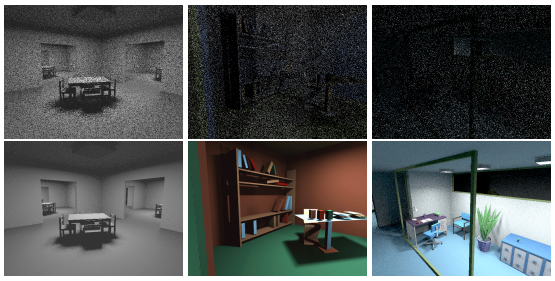

Figure 4: *Quality comparison at same number of paths in the ERW10, Ellipse, and Soda Hall Scenes. Top: Original method. Bottom: Using the estimated importance.*

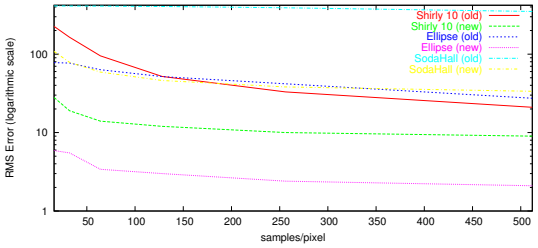

Figure 5: *RMS error to a master image over total rendering time: At the same rendering time, the new method shows up to ten times less RMS error.*

5. Instant Global Illumination in Complex and Highly Occluded Scenes

As just seen in the previous Section, importance sampling using our estimation step can result in much higher rendering performance for the proposed scenes at least for offline global illumination. Thus, it is an obvious next step to also apply this technique to interactive global illumination, which cannot currently handle such scenes.

Instant global illumination – as proposed by Wald et al. [18] uses an interactive ray tracing engine on a cluster of PCs to achieve interactive global illumination in a wide range of scenes. Since its original publication in [18], the method has been improved to feature higher image quality, programmable surface BRDFs, better scalability to dozens of PCs, and higher rendering performance [2].

5.1. Instant Global Illumination

In its core, instant global illumination builds on a variant of Kellers instant radiosity [7], in combination with interleaved sampling [8], and a filtering step. Instant global illumination (IGI) approximates the lighting in a scene by a set of virtual point light sources (VPLs) that are created by tracing a few "big" photons or particles from the light sources. These VPLs are then used to illuminate the scene just as with normal point light sources.

Due to the underlying fast ray tracing engine, IGI can efficiently handle even complex scenes of several million triangles. However, its efficiency depends to a large degree on the occlusion of a scene: As the performance is directly proportional to the number of VPLs used, only a small number of VPLs can be handled efficiently (in the order of 40 to 100).

In highly occluded scenes, most of these few VPLs will likely be located in parts of the scene where they do not contribute to the current image. In that case, many more VPLs than interactively affordable would have to be used to achieve a reasonably good coverage of the current view and obtain a good image quality. For example, consider rendering the ERW10 scene compared to rendering a single of its rooms, where 100 times as many VPLs would have to be used for the whole model in order to get the same quality as one would get for the single room.

Using our method, it should be possible to concentrate the few precious VPLs to the actually important rooms, and thus be able to render such scenes with good quality at interactive rates. However, integrating the new method into an interactive framework requires to solve new challenges due to the combination of an interactive setting with extremely low sample rates.

In principle, three problems have to be solved: First, the estimation and importance sampling steps have to be integrated into the distributed computing framework that IGI runs on. Though future IGI versions on new hardware platforms may not require this distribution framework, today this is unavoidable in order to achieve interactive performance.

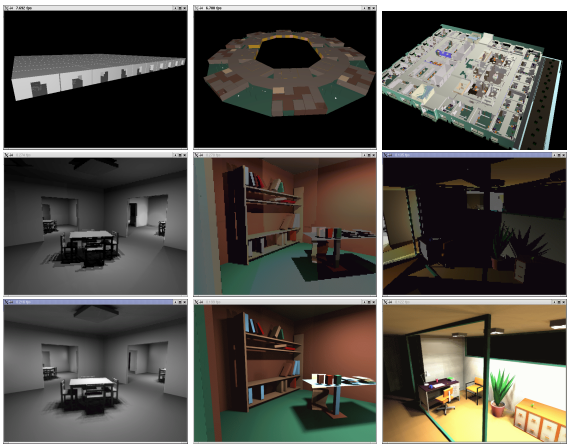

Figure 6: *Comparison of the image quality of the original instant global illumination method (middle) versus our new importance sampling technique (bottom row) in the ERW10 (80k triangles, 100 lights), Ellipse (19k triangles, 1,164 lights), and Soda Hall scenes (2.5M triangles, 23,256 lights), respectively, while running at the same frame rate. While the images computed with the old method are hardly recognizable the new method produces images of reasonably quality at the same frame rate. Note that the lower row uses even fewer VPLs due to the cost for estimation.*

Second, special care has to be taken when constructing the PDF in order to cope with the extremely small sampling rates. In particular, we have to make sure that light sources with small importance receive some samples at all. If, for example, one light source contributes 80 percent to the illumination in an image, giving it 16 out of a total of 20 VPLs would only leave 4 VPLs to represent all other light sources.

Finally – and most importantly – we have to consider temporal artifacts of our method. This is especially important due to the low sample rates that lead to slight errors that are hardly noticeable in a still image but which can lead to strong flickering between successive frames.

5.2. Distributed Implementation

The distribution framework used in the IGI system uses a tile-based dynamic load balancing scheme, which lets different clients compute different image tiles. In order to avoid seeing the tile borders in the final image, we have to make sure that all clients compute exactly the same image. For the original system, this was simple to achieve by synchronizing the seed values for the random number generators on all clients. Using the same random numbers, all clients would generate exactly the same VPLs, and thus would compute the same image.

In our new version, however, we also have to make sure that all clients use exactly the same PDF. Using a different PDF – even when using the same random numbers to sample it – would result in different VPLs, which in turn would be clearly visible as a tiling pattern on the image plane where each tile is illuminated differently.

In order to synchronize the PDFs, there are basically two choices: First, one could let all clients perform the full estimation step on the whole frame, and again use synchronized random numbers. Even though this would work, this approach would not scale and thus is not applicable in an interactive setting. Each client would have to shoot several hundred thousand rays just for the estimation step, which already exceeds the budget of rays it can shoot for a single frame.

In the second approach, each client performs the estimation only for its current tile. This perfectly fits the load balancing scheme as clients computing more tiles also perform more estimation computations. On the other side, it results in different PDFs on each client and requires a global synchronization. Due to the long latency of the underlying commodity network the clients cannot wait for the server to have received and synchronized the estimate from all clients for the current frame. However, if we tolerate one frame of latency for the estimate to be used, we have a simple and efficient solution. Each client performs the estimation for its current tile and sends the results to the server in compressed format. The server combines the different estimates and broadcasts the global estimate by attaching it to the first tile sent to each client for the next frame.

The latter approach obviously works only if the PDF for the next frame is not too different from the current frame. This is usually not a problem for walkthrough applications, but can result in an slight delay, e.g. when turning on a new light source or when newly entering a previously completely occluded room. In practice, however, the visual artifacts ususally remain small and tolerable.

5.3. Exploiting Temporal Coherence

For a typical walkthrough, successive frames usually do not differ too much. This temporal coherence can be exploited in several ways: First, we can smooth the importance estimate by combining the estimates from several successive frames. We currently do this by combining the old PDF with the current estimate using the weighed sum $P_{new} = \alpha P_{est} + (1-\alpha)P_{old}$. This allows us to use fewer estimate samples and thus to lower the estimation overhead.

Similarly, we can use the PDF from the last frame to also use importance sampling in the estimation process. This increases the reliability of the estimate in the interactive setting. However, we have to make sure that for the estimation step, each light source gets a minimum probability for being sampled. Otherwise, a light source that would once receive a zero PDF would never be sampled nor estimated again and would forever remain invisible. Though this eventually samples all light sources in the estimation step, the shadow rays are shot from the eye path towards the sampled sources, and thus will never touch geometry around occluded light sources.

Finally, another way of exploiting temporal coherence is to reuse eye rays that have to be traced anyway: As the estimate computed in the current tile will only be used in the next frame we can save time by not tracing separate primary rays for estimation *before* the rendering pass. Instead we reuse the primary rays already traced for rendering the current frame, thereby again reducing the estimation cost.

5.4. Avoiding Temporal Noise

The main problem to cope with in our approach is temporal noise, which may become visible as flickering of illumination. Even though both instant radiosity and our importance sampling step are unbiased in theory, the small number of samples (VPLs) affordable for interactive frame rates lead to a certain remaining error in the image. As this error is usually smoothly distributed over the entire image, it is often not noticeable in a still image. In an interactive setting however, two successive frames that are rendered with different VPLs may have their error in different parts of the image, resulting in visible flickering.

For the original system, this flickering could be controlled by using the same random numbers for successive frames, generating exactly the same VPLs for both frames. Using view-driven importance sampling, this is no longer possible, as any kind of user interaction – moving, turning, or interacting with a scene – will change the view importance for the next frame. As this view importance influences where the VPLs will be placed, any interaction will lead to “jumping” of the VPLs between frames.

In order to minimize this temporal flickering, we use the above-mentioned temporal smoothing of the PDF to minimize variations in the PDF. This however can also lead to an increased latency until a drastic change of a lights importance is taken into account.

Even more importantly, we make sure that the VPLs will be computed as coherent with the previous frame as possible. For example, the usual way of generating the VPLs would be to process all VPLs one after another by first sampling a light source (using our PDF), and then randomly placing a VPL on it. Then, however, it may easily happen that due to a change in the probability of another light source, a VPL will ’leave’ the light source it was placed on in the last frame. Now, even if the light source may receive another VPL, the new VPL will be generated by different random numbers, and thus be placed differently on the source. In practice, this leads to most VPLs jumping from frame to frame even with only small changes in the PDF.

In order to avoid this effect we have reversed the process. We first compute the number of VPLs that start at each light source using an error diffusion process to make up for rounding errors. Then for each light source that has received some VPLs, we start generating the VPLs with a random seed that is unique to the light source. Thus, the VPLs on a light source will always be placed in exactly the same way no matter how the PDFs of other light sources change. Also, if the number of VPLs on a light sources changes from n to m the first $\min(n,m)$ VPLs will remain the same, leading to drastically reduced flickering. Instead of processing all interleaving patterns independently, we perform this VPL distribution step for all lights of all interleaving patterns in order to maximize the average number of VPLs per light source.

However, if there remain many more active lights than the number of VPL paths that we compute for IGI, this trick will no longer work. In this case we can no longer expect VPLs to stay on any particular light sources if the PDFs change. This, however, could probably be solved by clustering nearby lights and distributing the VPLs according to these clusters in the same way as discussed above. This however has not yet been tested in practice.

5.5. Results and Discussion

Due to the interactive context it is hard to quantify the results of the new method in tables or present them as still images in a paper. The full impact of the improvements only become obvious when experienced interactively.

5.5.1. Temporal Artifacts

Temporal artifacts become most visible in the form of flickering and are mainly caused by the extremely small sampling rates used in an interactive context. As discussed above flickering is caused by some of the VPLs changing position between frames. The methods discussed in Section 5 use highly dependent solutions by placing the sampling in successive frames as consistent as possible. Essentially this tries to keep the remaining error as temporal consistent as possible. However, some change in the samples must be allowed in order to adapt the solution to the changing environment.

Another source of temporal artifacts is the occasional under-sampling of contributions by some light sources. This results in 'missing' illumination that may suddenly 'reappear' if the importance of the light source increases above the threshold. For example, imagine approaching a far away room that can only be seen through a small door: While far away this room may not receive any VPL and thus remains completely dark. When approaching its importance increases until it will receive its first VPL. This appears as if the light in this room had suddenly been "switched on".

The temporal artifacts can best be judged using the accompanying video, which shows several walkthroughs through our test scenes with both the original and with the new method. Though we usually cannot totally avoid all our method significantly improves the overall image quality of such walkthroughs, and usually already gives a good impression of the lighting in the scene. Note that in all of our experiments all illumination is fully recomputed every frame, allowing to arbitrarily and interactively change any lighting parameters, materials and geometry at any time.

5.5.2. Localization vs Non-localization

One obvious extension of our method would be to localize the importance sampling procedure by having each pixel choose its own subset of VPLs. This would allow to use different VPLs in different parts of the scene, and should reduce the above-mentioned undersampling artefacts. However, this localization would incur a high additional per-pixel cost for determining the VPLs, and would destroy the streaming framework of the IGI system which exploits the fact that all rays do exactly the same. Furthermore, it is unclear how the "jumping" of VPLs from image region to image region could be handled efficiently.

5.5.3. Estimation cost

One of the most obvious questions to quantify is the cost for the estimation step, which however is hard to determine: One obvious cost factor is the additional number of rays used for estimation. This, however, is very small compared to the bulk of the rays that is spent on shadow computations.

However, the rays used for the estimate are much less coherent than the shadow rays for the instant radiosity step and can be significantly more costly. Additionally, the estimation step requires other costly operations like samples light sources or path directions. There is also an additional cost for sending the estimates across the network, while the combination of the separate estimates on the server is negligible.

5.5.4. Overall Performance

Due to the discussed problems in exactly quantifying the impact of our method in detail, the best way of judging the improvements of our methods is to compare both methods side by side at the same frame rate. Therefore, we have taken the new method and have modified the quality parameters to achieve a reasonably good tradeoff for image quality versus rendering performance as it would typically be used with the original system in less complex scenes. For the comparison, we have then taken the original IGI system and adjusted its quality settings until the same frame rate was obtained.

The results of this side-by-side comparison can be seen in Figure 6: The image quality of the new method is generally much higher than with the original method. Whereas the original rendering quality is simply not tolerable at the given frame rate our method allows for image quality that is reasonably smooth. Although some artifacts are still visible, it nicely reproduces illumination features such as soft shadows that have not been present with the original method.

6. Conclusions and Future Work

In this paper, we have presented an efficient importance sampling technique for computing global illumination in complex and highly occluded scenes such as entire buildings. Using a cheap and purely view-importance driven estimation step our method can efficiently avoid sampling most occluded light sources. Thus, the sampling effort is concentrated almost exclusively on light sources actually contributing to an image. At the same frame rate, this results in a significantly improved image quality.

Applying our importance sampling technique to the instant global illumination system allows one for the first time to interactively explore entire buildings illuminated with highly complex geometry and thousands of light sources. It is important to note that no expensive or manual preprocessing of the scenes has been necessary. However, some temporal artifacts remain and become visible as flickering.

We expect that future refinement and localization of the importance estimate will reduce these temporal artifacts. More samples due to high-performance realtime ray tracing, possible even with hardware support, would also help. We strongly believe that once the necessary computational resources are commonly available, realtime global illumination will become a commodity for interactive 3D graphics, similar to the way textures became ubiquitous on todays graphics hardware.

Even today our method allows for interactive walkthroughs under full global illumination with reasonably good quality that would be sufficient for most practical demands. It is important to note that this is possible even with models that many other rendering algorithms can hardly render at all even without computing global illumination.

Acknowledgements

This work has been supported by Intel Corp. We also thank Philippe Bekaert and Daniel Menevaux for our test scenes, and Andreas Dietrich for his help and support for this paper.

References

1. Larry Aupperle and Pat Hanrahan. Importance and discrete three point transport. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 85–94, June 1993.
2. Carsten Benthin, Ingo Wald, and Philipp Slusallek. A Scalable Approach to Interactive Global Illumination. to be published at Eurographics 2003, 2003.
3. P. Hanrahan, D. Salzman, and L. Aupperle. A Rapid Hierarchical Radiosity Algorithm. In *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, pages 197–206, 1991.
4. H. Jensen and N. Christensen. Efficiently Rendering Shadows Using the Photon Map. In H. Santo, editor, *Edugraphics + Compugraphics Proceedings*, pages 285–291. GRASP-Graphic Science Promotions & Publications, 1995.
5. Henrik Wann Jensen. Global Illumination using Photon Maps. *Rendering Techniques 1996*, pages 21–30, 1996. (Proceedings of the 7th Eurographics Workshop on Rendering).
6. J. Kajiya. The Rendering Equation. In *Computer Graphics (Proceedings of SIGGRAPH)*, pages 143–150, 1986.
7. Alexander Keller. Instant Radiosity. *Computer Graphics*, pages 49–56, 1997. (Proceedings of ACM SIGGRAPH 1997).
8. Alexander Keller and Wolfgang Heidrich. Interleaved Sampling. *Rendering Techniques 2001*, pages 269–276, 2001. (Proceedings of the 12th Eurographics Workshop on Rendering).
9. Alexander Keller and Ingo Wald. Efficient Importance Sampling Techniques for the Photon Map. In *Vision Modelling and Visualization 2000*, pages 271–279, November 2000.
10. E. Lafortune and Y. Willems. Bidirectional Path Tracing. In *Proc. 3rd International Conference on Computational Graphics and Visualization Techniques (Compugraphics)*, pages 145–153, 1993.
11. Daniel Meneveaux, Kadi Bouatouch, Gilles Subrenat, and Philippe Blasi. Efficient Clustering and Visibility Calculation for Global Illumination. *AFRIGRAPH 2003*, pages 87–94, 2003. (Proceedings of AFRIGRAPH 2003).
12. Steven Parker, Peter Shirley, Yarden Livnat, Charles Hansen, and Peter Pike Sloan. Interactive Ray Tracing. In *Proceedings of Interactive 3D Graphics (I3D)*, pages 119–126, April 1999.
13. Matt Pharr, Craig Kolb, Reid Gershbein, and Pat Hanrahan. Rendering Complex Scenes with Memory-Coherent Ray Tracing. *Computer Graphics*, 31(Annual Conference Series):101–108, August 1997.
14. Peter Shirley, Changyaw Wang, and Kurt Zimmerman. Monte Carlo Techniques for Direct Lighting Calculations. *ACM Transactions on Graphics*, 15(1):1–36, 1996.
15. Brian Smits, James Arvo, and Donald Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. *Computer Graphics*, 28(Annual Conference Series):435–442, 1994.
16. E. Veach and L. Guibas. Bidirectional Estimators for Light Transport. In *Proc. 5th Eurographics Worshop on Rendering*, pages 147 – 161, Darmstadt, Germany, June 1994.
17. Ingo Wald, Carsten Benthin, Markus Wagner, and Philipp Slusallek. Interactive Rendering with Coherent Ray Tracing. *Computer Graphics Forum*, 20(3):153–164, 2001. (Proceedings of Eurographics 2001).
18. Ingo Wald, Thomas Kollig, Carsten Benthin, Alexander Keller, and Philipp Slusallek. Interactive Global Illumination using Fast Ray Tracing. *Rendering Techniques 2002*, pages 15–24, 2002. (Proceedings of the 13th Eurographics Workshop on Rendering).
19. Ingo Wald, Philipp Slusallek, and Carsten Benthin. Interactive Distributed Ray Tracing of Highly Complex Models. *Rendering Techniques 2001*, pages 274–285, 2001. (Proceedings of the 12th Eurographics Workshop on Rendering).
20. G. Ward. Adaptive Shadow Testing for Ray Tracing. In *2nd Eurographics Workshop on Rendering*, 1991.

Figure 7: *"Soda Hall", a 7 storey fully furnished building containing 2.5 million triangles and 23,256 light sources. a.) The entire building, simply shaded, b.) Overview of a single floor c.) and d.) Inside, with full global illumination. With our proposed technique, the right images run at 2-3 frames per second on 22 CPUs allowing to interactively explore the entire building with full global illumination.*

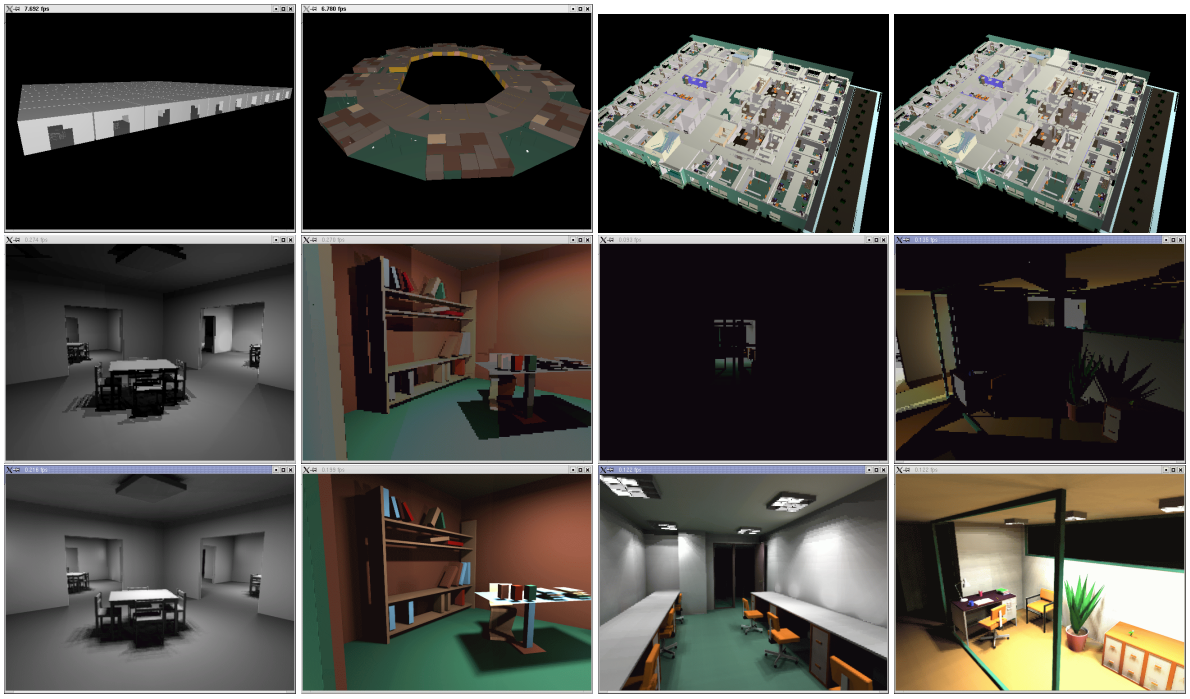

Figure 8: *Examples for complex and highly occluded scenes. Top row, from left to right: Overview of the entire scenes: "Shirley 10" (80k triangles, 100 lights), "Ellipse" (19k triangles, 1,164 lights), and "Soda Hall" (2.5M triangles, 23,256 lights), respectively. Comparison of the image quality of the original instant global illumination method (top row) versus our new importance sampling technique (bottom row) in the ERW10, Ellipse, and Soda Hall (two views) scenes, respectively. Due to the cost for estimation, the lower row uses even less VPLs than the upper row (usually about 50%). While the images computed with the old method are hardly recognizable the new method produces images of reasonably quality at the same frame rate. Especially note the quality of the shadows.*