Volume xx (200y), Number z, pp. 1–9

Progressive Point-Light-Based Global Illumination

H. Dammertz[1] and A. Keller[2] and H. P. A. Lensch[3]

[1] holger.dammertz@uni-ulm.de Ulm University, Germany
[2] alex@mental.com, mental images GmbH, Germany
[3] hendrik.lensch@uni-ulm.de Ulm University, Germany

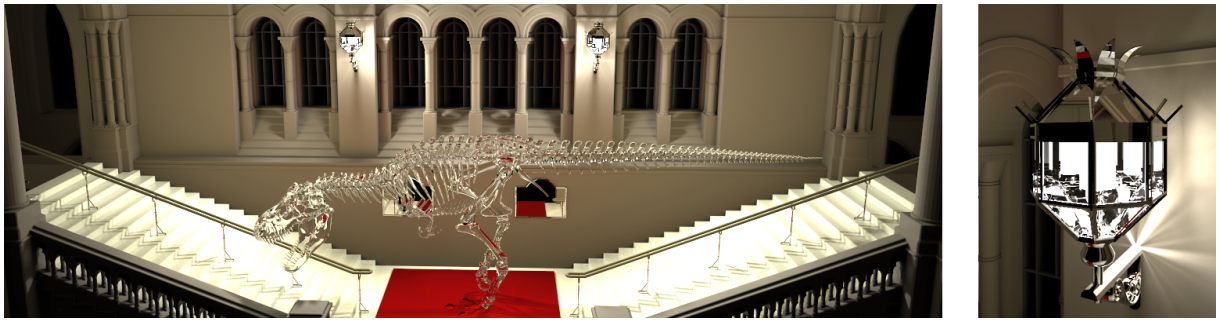


Figure 1: *Our progressive rendering system is able to handle a wide range of complex light paths efficiently by combining three approaches. The shown scene contains diffuse and specular objects that are illuminated only by two lamps enclosed in glass. Our system not only handles the bright areas well but allows for smooth indirect illumination from this kind of light sources without the need for parameter tuning.*

Abstract

We present a physically-based progressive global illumination system that is capable of simulating complex lighting situations robustly by efficiently using both light and eye paths. Specifically, we combine three distinct algorithms: point-light-based illumination which produces low-noise approximations for diffuse inter-reflections, specular gathering for glossy and singular effects, and a caustic histogram method for the remaining light paths. The combined system efficiently renders low-noise production quality images with indirect illumination from arbitrary light sources including inter-reflections from caustics and allows for simulating depth of field and dispersion effects. Our system computes progressive approximations by continuously refining the solution using a constant memory footprint without the need of pre-computations or optimizing parameters beforehand.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism: Color, shading, shadowing, and texture—

1. Introduction

In this paper we present a progressive bidirectional physically-based rendering system that improves the perceived image quality in many situations. For this we combine three techniques (virtual point light sources, caustic histograms and specular gathering) in a robust way. Each technique contributes a distinct part to the final image and together they are able to simulate for example caustics, specular surfaces, and depth of field effects.

The benefit of the combination is that the techniques can take advantage of each other by reusing information where appropriate and that the advantages of each technique are retained:

- virtual point light sources (VPL): smooth indirect illumination
- caustic histograms (CH): crisp, efficient caustics
- specular gathering (SG): robust handling of glossy and singular materials,

The caustic histogram method collects photons in bins that are created for each of the progressively sampled eye paths. Although we include approximating techniques and thus accept a certain small bias in the solution, the progres-

sive rendering algorithm consistently converges to a unique solution, especially in complex illumination situations.

We argue that a fully progressive system allows for much easier handling of the renderer by artists because the choice of parameters such as number of samples and photons does not change the final image quality (they may affect only rendering time). In addition, the proposed system decouples memory consumption and image quality (i.e. higher quality renderings do not require more memory at runtime). Compared to many previous approaches our system renders low-noise production quality images with indirect illumination from arbitrary light sources including inter-reflections from caustics and allows for simulating depth of field and dispersion effects.

2. Related Work

Physically-based rendering is a sub-problem of image synthesis, which aims to simulate light transport on a computer in a physically correct way. The problem has been studied in great detail [PH04, DBB06] and as it stands, ray tracing is the only method that allows to solve complex illumination problems consistently. Even though ray tracing performance has been improved greatly in the recent years [Shi06], only few of the achievements directly help Monte Carlo based global illumination algorithms [BWB08, DHK08]. Several hundreds of million paths need to be ray traced for a final image and thus Monte Carlo and quasi-Monte Carlo rendering algorithms have a very high computational demand and long rendering times. Unbiased Monte Carlo-based rendering systems are progressive by nature and thus have the advantage of robustly handling even complex situations with the only limitation being the rendering time. The disadvantage is that even in simple situations unbiased Monte Carlo algorithms have a high variance perceivable as noise in the image.

Kajiya [Kaj86] introduced Monte Carlo path tracing algorithms to computer graphics. Since then, a lot of research headed to improve the efficiency of the basic algorithm. The family of bidirectional path tracing algorithms using multiple importance sampling [LW93, VG94] and the Metropolis light transport algorithm [VG97] belong to the most powerful algorithms to date. While these algorithms are unbiased and can deal with complex lighting situations, they suffer from variance, which becomes visible as noise in images. Our system approaches the global illumination from both light and eye paths using a Monte Carlo technique, however, it reduces variance through the use of correlated point light sources for diffuse inter-reflections. The basic approach of splitting the final solution to be computed by different algorithms was also described in [CRMT91].

The Instant Radiosity algorithm [Kel97] uses point light sources and graphics hardware to quickly compute a global illumination solution for diffuse scenes. Since then, point-light-based global illumination has proven to be a useful and efficient way to approximate diffuse inter-reflection in real time systems [BWS03] and preview systems [HPB07]. In [SIMP07] an extension to the Instant Radiosity algorithm is presented that uses Metropolis sampling to increase efficiency. The use of VPLs is at the core of our rendering system, however, we extend it to simulate non-diffuse light paths as well.

Photon mapping [Jen96] has been introduced as a solution to the problem that there exist paths that cannot be efficiently sampled by any unbiased technique. With the recent improvements [HOJ08], many shortcomings of the original method have been removed. In our approach we remove the remaining memory bound on the number of eye path vertices that need to be stored, which for example allows one to simulate anti-aliased depth of field with any sampling rate required. These problems were also recently addressed in [HJ09]. Other recent research focuses on GPU based methods to accelerate the illumination computation [WWZ*09].

The Lightcuts rendering framework [WFA*05, WABG06] is a powerful approach to the many-lights problem, which is entirely based on point-light sources. Due to the ability to deal with an enormous number of point light sources, glossy effects can be handled, however, the system cannot simulate caustics and is not progressive. As our approach is progressive there is no limitation on the number of light sources and even the simulation of caustics is integrated.

3. System Overview

Our physically-based rendering system is designed to produce production quality images of 3D scenes with global illumination in complex situations without the need for extensive parameter tweaking. We therefore assume physically plausible input: for example light sources are modeled as geometry that is part of the scene, glass objects always have a thickness, and surface shaders need to be energy conserving. Progressiveness enables the user to continue image computation until a satisfactory result is achieved. There is neither the need to restart computations nor to discard intermediate results that allow for previews of an illumination situation after a short amount of time.

The goal of our system is to provide a fast and smoother global illumination solution as compared to approaches using multiple importance sampling [VG95, Vea97, VG97] without sacrificing too much quality and flexibility. We achieve this goal by combining three techniques, each of which simulates a disjoint subset of path space in such a way that the techniques benefit from each other. For ease of reading, we apply Heckbert's [Hec90] notation to identify path subspaces: E is the eye or camera, D denotes a diffuse bounce (in Section 4 we define what diffuse means for our rendering system), S is a non-diffuse (specular) bounce, and L represents the light sources.

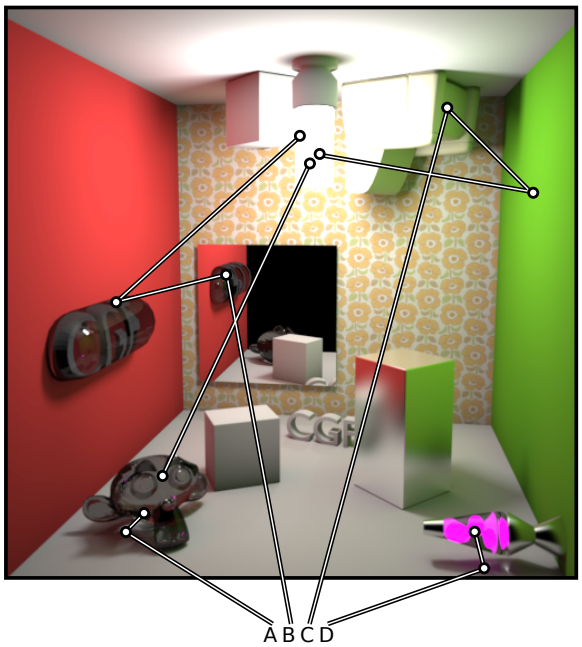


Figure 2: *Illustration of selected light paths that we are able to simulate robustly: 'A' is a classic caustic path (EDSSL), 'B' shows the mirror image of a diffuse object behind a glass object (ESSDSL), 'C' is diffuse indirect illumination from the green wall to the sofa (EDDL), and 'D' shows a complex shaped light source (lava lamp) behind glass illuminating the scene (EDSL). Also note the caustic generated by the monkey head in the mirror (ESDSSL).*

Figure 2 illustrates the light transport paths we are able to simulate.

The techniques and the simulated light paths in our system are:

- EDL, $EDD(S|D)^*L$**:** is handled by the point-light-based algorithm (Section 5.1). These paths are directly visible diffuse surfaces that are directly or indirectly illuminated by another diffuse surface.
- $EDS(S|D)^*L$**:** is created by the caustic histogram method (Section 5.3). This is the classic caustic situation where a diffuse surface is illuminated indirectly from a specular surface.
- $ES^+D^+(S|D)^*L$**:** is handled by specular gathering (Section 5.2) and uses the results of the previous two methods. This are light paths of directly or indirectly lit surfaces seen through a specular reflection or refraction.
- ES^*L**:** is directly evaluated as in a path tracer [DBB06]. Direct connection of the eye with a light source using only specular bounces.

Looking at the path notation it can be seen that all these contributions are distinct and two methods can never generate a contribution for the same path. All four techniques together can generate every possible path $E(D|S)^*L$. The final solution is simply obtained by summation.

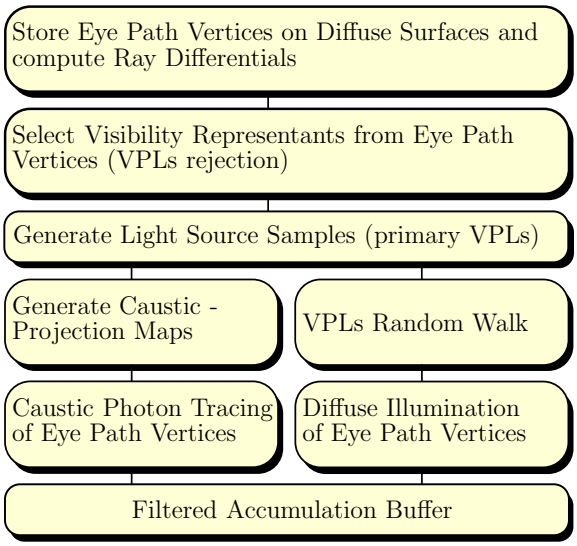


Figure 3: *This figure describes the order of the basic operations our rendering system performs each rendering pass.*

Figure 3 shows the steps our rendering system performs in each rendering pass. A rendering pass is one iteration of computation that is finally added to the accumulation buffer containing the progressively refined result image. In the following sections each of the steps is explained in detail.

3.1. Eye Path Generation

The first step of our rendering system is the generation of eye path vertices. This is illustrated in Figure 4 (a). The eye paths are created by a random walk starting from the camera and are terminated via Russian roulette or when they hit a diffuse surface. The vertices are stored only on diffuse surfaces. These points are similar to the gather points in the Lightcuts [WABG06] algorithm (but we store only the end points of a path) or the hitpoints stored in the progressive photon map algorithm [HOJ08]. The main difference is that we create a new set of eye path vertices each rendering pass instead of keeping them for the whole image generation process. This allows for progressive refinement of spatial anti-aliasing and complex specular effects. In the standard setting one eye path is started per pixel for each rendering pass. When such a path hits a light source (either directly or via an arbitrary number of specular bounces: ES^*L) the contribution is directly added into the accumulation buffer but the path is not necessarily terminated as the surface of the light source might also reflect light. Using the generated eye path vertices we will gather diffuse illumination (Section 5.1), specular contributions (Section 5.2) and caustics (Section 5.3).

4. Material Model

While general production renderers for the movie industry can use arbitrary complex programs to define surface color and reflection behavior [DH05, Kes08], physically-based systems are more restricted in their choice of reflection functions [PH04]. Many practical algorithms exploit the fact that BRDFs are not only energy-conserving, but also provide an efficient sampling function.

A wealth of surface reflection models has been developed in computer graphics that provide most of the required functionality [Bli77, War92, AS00]. These reflection models allow one to describe a wide variety of basic surface types. For more complex surface behavior usually several reflection models are combined to a single layered material [PH04].

Systems using multiple importance sampling [PH04] require the BRDF model to satisfy the Helmholtz reciprocity condition in order to obtain consistent results. Since we partitioned path space and use only one technique on each partition, this condition no longer needs to be fulfilled, allowing one to use more convenient approaches like e.g. the halfway vector disc model [EBJ*06], while still enjoying convergence to a unique solution.

In order to determine the partition of the path space, we need to decide when a reflection is almost diffuse. In principle our system can use any physically-based layered BRDF model for which a parameter κ can be assigned to each layer expressing how diffuse it is. This parameter is assumed to be normalized such that for $\kappa = 0$ the layer is perfect Lambertian and for $\kappa = 1$ the surface is a perfect mirror. During rendering each sample evaluates only one material layer that is selected by random sampling. We use the threshold of $\kappa = 0.2$ for classifying a material layer as diffuse or specular.

For example, in all renderings we use the Blinn-Phong model [Bli77] with Phong-exponent $k = 1024\kappa + 1$. Layered materials such as simple metal or glass (one reflecting, one transmitting layer) and more complex materials like coated plastic can be constructed easily using the Fresnel term or one of the approximations [Sch94] for weighted sampling of material layers. For transmissive materials the direction is selected according to Snell's law but the width of the lobes is controlled as before, as this allows for diffuse and imperfect glass.

5. Bidirectional Global Illumination

Based on the material properties we partition the path space. In the following we describe how the different subsets are sampled.

5.1. Point-Light-Based Diffuse Illumination

We approximate the diffuse illumination by virtual point light sources (VPL) similar to the Instant Radiosity algorithm [Kel97]. In order to be consistent with existing literature we use the term VPL (Virtual Point Lightsource) for a light path vertex even though we prefer the term DIAL (Diffuse Illumination-Approximating Light) in the context of our rendering system. Note that the notion of diffuse illumination not only includes perfectly Lambertian surfaces, but also slightly glossy surfaces as specified in Section 4.

The first step in the Instant Radiosity algorithm generates samples on the light sources. As our scenes contain arbitrarily shaped light sources modeled as triangles, we first compute a PDF (probability density function) according to the area and intensity of each emitting triangle. This PDF is created once and can be reused for all subsequent rendering passes. Using three random numbers, a position on the light sources (stored as primary VPL) is sampled. Then a random walk terminated by Russian roulette is performed in the manner of a particle tracer, where on each diffuse surface another VPL is stored. Unlike eye paths, the light paths are not terminated on the first diffuse hit. This is illustrated in Figure 4 (b). These VPLs represent a point-wise approximation of the diffuse inter-reflection. In contrast to [Kel97], the contribution of each VPL is then accumulated by sampling the direct visibility of the VPLs to each pixel via the stored eye path vertices, i.e. possibly incorporating even multiple specular bounces (see next section). Figure 4 (c) illustrates this. This accumulation is inherently progressive as each series of VPLs, generated by one random walk, is independent of the other random walks and the final image is independent of the number of VPLs created per rendering pass.

Stochastic Culling of VPLs. As we want to be able to efficiently handle complex scenes, we use an additional optimization to discard VPLs that are likely to have no contribution. For this purpose we choose a stratified subset of 256 of the eye path vertices and check the visibility from these locations prior to VPL storage. This can be either done based on an acceptance probability to not introduce an additional bias, or fully deterministic. To guarantee a good coverage and a changing subset for each rendering pass we use the Halton sequence over the image plane for the selection [KW00]. The culling is recreated each frame with a new set of samples to guarantee that even unlikely paths that might contribute significantly are not missed during the progressive rendering.

5.2. Glossy and Singular Effects

Glossy and singular effects need to be simulated when the BRDF is not sufficiently diffuse (see Section 4). The idea of specular gathering is to sample the BRDF according to its probability density function by extending the eye path. This extension of the eye path is repeated until either it is terminated by Russian roulette or the path hits a diffuse surface. At this point the illumination is computed by using the point light sources and the caustic histogram (see next section). So, per rendering pass we gather the illumination for

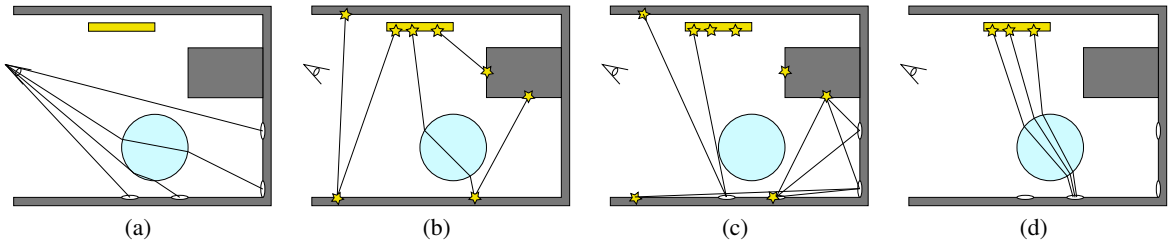


Figure 4: *(a): Generation of the eye path vertices by performing a random walk into the scene starting from the eye. (b): Generation of the VPLs by a random walk starting from light sources. (c): Evaluation of the VPLs to compute an approximation of the diffuse illumination at the eye path vertices. (d): Accumulation of caustic photons into eye path vertices.*

specular eye paths by using the current global illumination approximation given by the two other techniques. As this approximation is recomputed in each pass, the specular part becomes progressively refined, too.

5.3. Caustic Histogram Method

The final technique of our rendering system is the histogram method that provides the caustic paths. It is a two step method that is performed for each rendering pass. The two steps are illustrated in in Figure 4 (a) and Figure 4 (d). In the first step, the eye path vertices are created. Each eye path vertex has an associated bin size. To estimate the bin size we trace ray differentials [Ige99] starting with half the pixel size. This provides a good estimate of projected pixel footprint even after specular bounces. This step produces the bins that are filled with photons in the second step. The bins are stored in a *k*d-tree. Now for each primary VPL (which are just samples on the light source) we trace photon random walks into the scene. If they do not hit a specular material they are discarded, otherwise they are continued until a diffuse material is found (or they are terminated by Russian roulette). The contribution to the image is computed by adding the value to the accumulation buffer using the associated pixel from each bin. Each photon is only accumulated into a single bin each bounce by choosing the closest. This way bins do never overlap and due to the euclidean distance calculation the bins are sphere shaped where they do not overlap.

This procedure is illustrated in Figure 4 (d) for a glass sphere. In our implementation we trace photons until we have accumulated a given number (100000 in all images) for each rendering pass. Using the previously computed VPL as emitting light sources has the advantage of not requiring a new set of light source samples. The number 100000 gives in our case a good balance between the caustics and the other parts of the illumination solution. Changing this number does not change the converged result, it might only change the convergence speed of caustics vs. the rest of the global illumination solution.

This method is conceptually similar to the algorithms described in [DLW93,SW00a,HOJ08]. However, first, we only use the caustic histogram for the $EDS(S|D)^*L$ paths, and second, constant radius per eye path vertex is used for the bins based on the projected pixel size, which speeds up the photon collection. Furthermore, we recreate the bins in each rendering pass and thus in addition are able to support progressive per pixel anti-aliasing, glossy surfaces, depth of field, and motion blur. The fixed radius selection may of course introduce the usual photon mapping artifacts and introduces a bias, but as the (projected) bin size is not larger than half the pixel size the error is not larger than the error introduced by image space filters [SW00a]. The recreation has also been recently developed in [HJ09].

Figure 5: *Illustration of four of the projection maps of the scene shown in Figure 10. The third map is from a VPL inside the lava lamp.*

Projection Maps. Even for arbitrarily complex light sources only relatively few VPLs are generated per rendering pass. As we emit caustic photons from these VPLs we can make efficient use of a projection map per VPL as opposed to previous approaches of using projection maps with photon emission [Jen01].

Projection maps are a pre-computation to direct caustic photons only in areas where glossy/specular objects are. This increases the efficiency in simple scenes (with few glossy objects) significantly without slowing down complex scenes.

We create the projection map as a conservative projection of all triangles classified as specular onto the hemisphere of each VPL and then use correctly weighted sampling to reject photon rays that would surely hit only diffuse surfaces. In or-

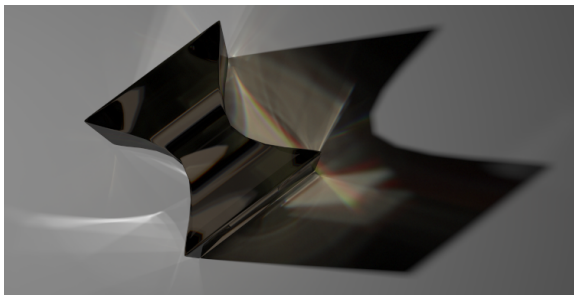


Figure 6: *(745 × 380 0.8 hours) On demand spectral rendering allows to integrate dispersion into an RGB based render.*

der to efficiently splat the spherical triangles we approximate them by the bounding box of the projected vertices. Figure 5 shows some projection maps created in the Box scene. We used a resolution of 128×256 for each projection map. During rendering the generation of projection maps was between 2% and 5% of the total time spent on the caustic histogram method. The speed-up achieved is linear to the coverage of the hemisphere, saving 85.2% of the photon rays in the Box scene, corresponding to 65.3% of the time required to compute the caustics and 4.6% of the total render time.

Implementation Details. In practical applications the number of eye path vertices per rendering pass is larger than our chosen number of photons per pass. Since our accumulation of photons into vertices is redone every pass we can reverse the order and store the photons instead of the eye path vertices as in the original photon map algorithm [Jen96]. This does not change the result of the computation but may be more memory efficient for high resolution images.

6. Lazy Spectral Rendering

For efficiency reasons our system is based on RGB rendering. Yet it is possible to consistently integrate an approximation for dispersion effects into our rendering system by lazy spectral rendering. This works by using the normal RGB transport until a ray hits a dispersive surface (e.g. glass). At this point the ray is assigned a randomly sampled wavelength from the original RGB color. In addition, we further assign a novel RGB color corresponding to the sampled wavelength. Once a ray has an assigned wavelength it will keep it until the ray terminates.

The wavelength assignment is done for all rays generated in the system: eye-paths, photon-paths and the random walks to generate the VPLs. This allows for the usual color caustics (photon paths), the chromatic aberration seen through refractive objects (eye-paths) and correct indirect illumination from colored caustics (VPLs). Figure 6 shows a rendering with a dispersive glass object.

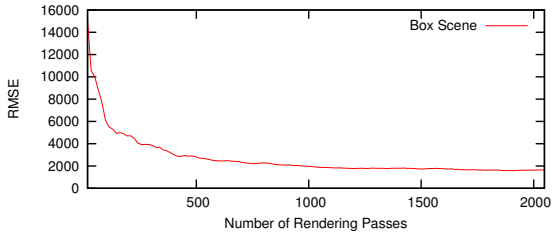


Figure 7: *This graph shows the errors of the images (computed at* 1024×1024*) shown in Figure 10 from* 16 *to* 2048 *rendering passes in steps of* 16 *compared to a converged image with* 11456 *rendering passes. The steep descent at the beginning is because the indirect illumination is quickly approximated by the point light sources.*

7. Results

In this section we discuss the results of our rendering system. All images in this paper were computed using the same default parameter settings of 16 VPL paths, 1 eye path per pixel, and 100000 caustic paths per rendering pass. All measurements were done on a Core(TM)2 Quad Q9550 @ 2.83GHz.

Figure 11 shows the statistics for the different scenes used to present different aspects of our system. Figure 10 shows the image generation process with the inverse difference image scaled by 8 compared to a converged image with 11456 render passes. In Figure 7 you can find the associated graph showing the mean square error.

Diffuse Illumination. The diffuse illumination (direct and indirect) is approximated by the use of VPLs and thus very robust. This method was already discussed in many publications and is known to produce good results with a sufficient number of point light sources. As our system is progressive, the only limiting factor for the number of point light sources is rendering time. Figure 8 shows an equal time comparison of a bidirectional path tracer (BDPT) and point light based global illumination. The difference is most noticeable in the darker areas where the detail is still obstructed by noise in the BDPT while the VPLs approximation already shows each feature clearly. This robustness of indirect illumination in darker areas can also be seen in Figure 1 where the far left and right sides are only illuminated by indirect light. Note that the use of point lights for diffuse illumination also allows for efficient handling of diffuse transparent objects like curtains by creating VPLs also on the exit points (two sided) during the random walk.

Glossy Effects. Figure 9 shows a simple single layered material where κ is varied from 0 to 0.9. The first three spheres in the last row have a $\kappa \in (0, 0.1, 0.199)$ and are thus illuminated by VPLs directly (the specular threshold is $\kappa \geq 0.2$). The remaining spheres are illuminated by specular gathering, i.e. by extending the eye paths. Note, how these two techniques generate a consistent transition.

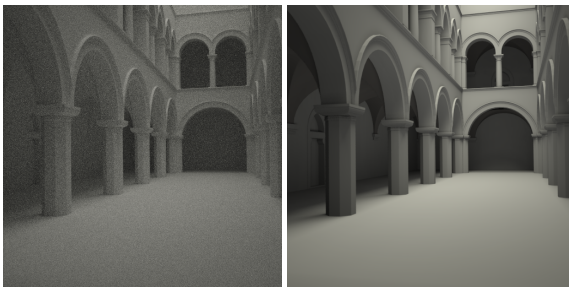


Figure 8: *Equal time comparison (*512×512 *3 minutes) of a BDPT (left) and the point light based renderer (right) in a simple diffuse scene. The noise in the BDPT solution is especially noticeable in the dark areas where none of the details are yet visible while in the other image the door and ceiling is clearly visible on the left side.*

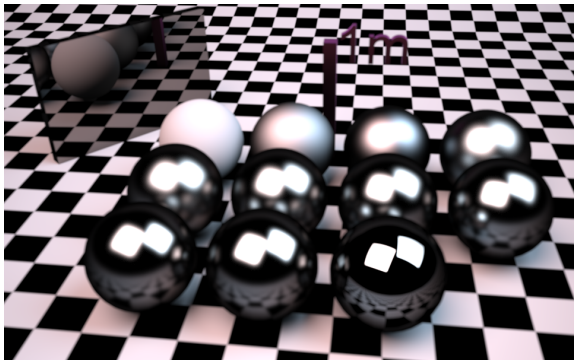


Figure 9: *This image shows the smoothness of changing from VPLs based illumination for diffuse illumination (*$\kappa < 2$*, Section 5.1) to specular surfaces (*$\kappa \geq 2$*, Section 5.2). Due to the progressive rendering both methods converge to a final solution where no difference is noticeable without the need to set the correct number of samples or number of point light sources.*

The top right image in Figure 11 shows a scene with many glossy objects and high variance situations where glossy objects are reflected in other glossy objects. In addition, the scene features depth of field. The illumination comes from a sky model only through the two small windows.

Caustics and other *SDS* paths. Caustics and *SDS* paths are efficiently treated by the caustic histogram method using projection maps (Figure 1 and Figure 2). The *SDS* paths are especially difficult in BDPT-based renderers and take a long time to converge even with Metropolis sampling. Without them the ground below the feet of the glass dinosaur in Figure 1 would be black. Section 5.3 further demonstrates how spectral effects can be integrated in our rendering system. The effect of spectral rendering is most noticeable in caustic areas, however, the presented method integrates well with the other proposed parts of the rendering system.

Depth of Field. Since we create a new set of eye path vertices in each rendering pass we can easily simulate e.g. a thin lens camera that renders depth of field effects. As our system is progressive we do not need to adjust any of the parameters for the illumination computation and, in many cases, the number of rendering passes needed for the illumination to converge is sufficient to robustly estimate depth of field effects (Figure 11).

Comparison to Progressive Photon Mapping and MLT Figures 12 and 13 show a direct equal-time comparison of our proposed rendering method to progressive photon mapping and to metropolis light transport. For the first image we used a render time of 8 minutes. For the second image, 15 minutes were used. All images were created at a resolution of 512×512. Figure 12 is a simple box setting with diffuse colored walls. As it can be seen in the images our method produces much smoother results while still retaining a high quality caustic. Figure 13 shows more complex light paths through a glass object. The object is placed on a large diffuse plane. Here as well, our method produces much smoother results without sacrificing image detail. The progressive photon mapping algorithm still exhibits some low-frequency noise because it cannot gather enough photons within the given render time. In contrast, the metropolis algorithm still has considerable high-frequency noise. Note that also progressive per pixel anti-aliasing is not possible in the progressive photon map algorithm since the gather-points are fixed for the whole image generation process. For the metropolis comparison we used the open source luxrender v0.6rc5 as it contains a well optimized implementation. This explains also the slight difference in image appearence as this rendering system handles light sources and especially materials a little different from our system. The last row shows the difference image to the path traced solution. It can be observed that each method has unique areas where the difference is still quite high after the 15 minutes of computation time. The progressive photon map shows especially at the geometry silhouettes the lack of anti-aliasing but also quite high differences in the area where few photons are transmitted and thus the gathering radius is still large. This is the same area where the metropolis also has some higher noise. In contrast to this our algorithm spreads the error more evenly and shows an almost noise free error image.

8. Limitations

Indirect caustics (EDS^+D^+L) are created from non-primary VPLs. This works well when diffuse surfaces are brightly lit, but in many scenes this increases the variance too drastically for the final visible effect. So we chose to make them artist-controllable instead of enabling them by default. Furthermore, the depth of indirect caustics can be selected prior to rendering.

The bias of bounding the geometric term in the Instant

Radiosity method can be removed by the method proposed in [KK04]. The clamped contribution of point lights is compensated by gathering the missed illumination through additional rays. This comes at the cost of increasing the variance, especially in corners for an effect that is only seen in few scenes. We again made this an artist-controllable option.

Using a material model that can not be split easily into a diffuse and a specular part (for example when using measured BRDFs like in [MPBM03]) might increase the variance and thus the time to a high quality image. One could use a simple heuristic to classify a BRDF as more diffuse or more specular but a wrong choice results in longer computation times. Note that this does not change the result image, only the needed time to compute it.

While our proposed rendering system is progressive by design, it is not yet adaptive in any way (except that in highly glossy scenes automatically fewer VPLs are stored). Especially in diffuse areas the resulting image quickly converges to an acceptable solution while in areas of *SDS* paths there is still noticeable variance. By intuition one would like to sample only in areas of high variance. But this would remove some of the advantage of a real progressive renderer as it cannot be guaranteed that the undersampled region is no area of higher variance (for example light through a key hole) that was not yet sampled. Of course knowledge about a given scene allows an artist to increase convergence speed by tuning the number of point light sources or caustic photons per rendering pass accordingly.

9. Conclusion

We showed how a combination of three distinct techniques based on point light sources allows us to compute a global illumination solution including caustics, specular surfaces, depth of field, and dispersion effects in a single unified rendering system. Each of the techniques - VPLs, specular gathering and the novel caustic histograms - efficiently works with the other two in combination. The progressive nature of our system allows the artist to start a rendering and be sure that it will converge to a unique solution independent of the parameter choice. At least in theory our system would produce the same image even if just a single VPL path or a single photon path would be generated per rendering pass. Additionally, our system uses only a constant amount of memory during rendering independent of the final image quality.

Future Work. Subsurface scattering can be integrated in our system by allowing eye path vertices inside solid objects and tracing the photons also through these objects. Additionally, participating media using the VPLs can be integrated [RSK06]. Adding motion blur requires to sample just another dimension in our progressive renderer [CFLB06]. A desirable feature for the whole rendering system would be to automatically balance the three parameters (number of VPLs, number of specular gathering rays, and the number of photons for the histogram) for optimal convergence speed with respect to a given scene.

10. Acknowledgments

The first author would like to thank mental images GmbH for support and funding of this research. The first author would also like to thank Johannes Hanika and Matthias Raab for endless discussions about physically based rendering.

References

[AS00] ASHIKHMIN M., SHIRLEY P.: An anisotropic Phong BRDF model. *Journal of Graphics Tools 5*, 2 (2000), 25–32.

[Bli77] BLINN J.: Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.* (1977), 192–198.

[BWB08] BOULOS S., WALD I., BENTHIN C.: Adaptive ray packet reordering. *IEEE Symposium on Interactive Ray Tracing* (Aug. 2008), 131–138.

[BWS03] BENTHIN C., WALD I., SLUSALLEK P.: A Scalable Approach to Interactive Global Illumination. *Computer Graphics Forum 22*, 3 (2003), 621–630. (Proceedings of Eurographics).

[CFLB06] CHRISTENSEN P., FONG J., LAUR D., BATALI D.: Ray tracing for the movie 'Cars'. In *Proc. 2006 IEEE Symposium on Interactive Ray Tracing* (2006), pp. 73–78.

[CRMT91] CHEN S. E., RUSHMEIER H. E., MILLER G., TURNER D.: A progressive multi-pass method for global illumination. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (1991), pp. 165–174.

[DBB06] DUTRÉ P., BALA K., BEKAERT P.: *Advanced Global Illumination*. AK Peters, Ltd., 2006.

[DH05] DRIEMEYER T., HERKEN R.: *Programming Mental Ray (mental ray®Handbooks)*. Springer-Verlag New York, Inc., 2005.

[DHK08] DAMMERTZ H., HANIKA J., KELLER A.: Shallow bounding volume hierarchies for fast SIMD ray tracing of incoherent rays. In *Computer Graphics Forum (Proc. 19th Eurographics Symposium on Rendering)* (2008), pp. 1225–1234.

[DLW93] DUTRÉ P., LAFORTUNE E., WILLEMS Y.: Monte carlo light tracing with direct computation of pixel intensities. In *Proceedings of Compugraphics '93* (1993), pp. 128–137.

[EBJ*06] EDWARDS D., BOULOS S., JOHNSON J., SHIRLEY P., ASHIKHMIN M., STARK M., WYMAN C.: The halfway vector disk for BRDF modeling. *ACM Transactions on Graphics 25*, 1 (2006), 1–18.

[Hec90] HECKBERT P.: Adaptive radiosity textures for bidirectional ray tracing. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (1990), pp. 145–154.

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers* (2009), pp. 1–8.

[HOJ08] HACHISUKA T., OGAKI S., JENSEN H.: Progressive photon mapping. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers* (2008), pp. 1–8.

[HPB07] HAŠAN M., PELLACINI F., BALA K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. 26*, 3 (2007), 26.

[Ige99] IGEHY H.: Tracing ray differentials. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), pp. 179–186.

[Jen96] JENSEN H.: Global illumination using photon maps. In *Rendering Techniques '96 (Proc. of the Seventh Eurographics Workshop on Rendering)* (1996), pp. 21–30.

[Jen01] JENSEN H.: *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., 2001.

[Kaj86] KAJIYA J.: The rendering equation. *SIGGRAPH Comput. Graph. 20*, 4 (1986), 143–150.

[Kel97] KELLER A.: Instant radiosity. *ACM Transactions on Graphics (Proc. SIGGRAPH 1997)* (1997), 49–56.

[Kes08] KESSON M.: Pixar's renderman. In *SIGGRAPH Asia '08: ACM SIGGRAPH ASIA 2008 courses* (2008), pp. 1–138.

[KK04] KOLLIG T., KELLER A.: Illumination in the presence of weak singularities. *Monte Carlo and Quasi-Monte Carlo Methods 2004* (2004), 245–257.

[KW00] KELLER A., WALD I.: Efficient importance sampling techniques for the photon map. In *Proceedings of the Vision Modeling and Visualization Conference* (2000), pp. 271–279.

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques* (1993), pp. 145–153.

[MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3 (July 2003), 759–769.

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Elsevier, 2004.

[RSK06] RAAB M., SEIBERT D., KELLER A.: Unbiased Global Illumination with Participating Media. In *Proc. Monte Carlo and Quasi-Monte Carlo Methods 2006*. Springer, 2006, pp. 591–605.

[Sch94] SCHLICK C.: An inexpensive BRDF model for physically-based rendering. *Computer Graphics Forum 13* (1994), 233–246.

[Shi06] SHIRLEY P.: State of the art in interactive ray tracing. In *ACM SIGGRAPH 2006 Courses* (2006).

[SIMP07] SEGOVIA B., IEHL J.-C., MITANCHEY R., PÉROCHE B.: Metropolis Instant Radiosity. *Computer Graphics Forum 26*, 3 (2007), 425–434.

[SW00] SUYKENS F., WILLEMS Y.: Adaptive Filtering for Progressive Monte Carlo Image Rendering. In *Eighth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG 2000)* (Plzen, Czech Republic, 2000).

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997.

[VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Rendering Techniques '94 (Proc. of the Fifth Eurographics Workshop on Rendering)* (1994), pp. 147 – 161.

[VG95] VEACH E., GUIBAS L.: Optimally compining sampling techniques for Monte Carlo rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH 1995)* (1995), 419–428.

[VG97] VEACH E., GUIBAS L.: Metropolis light transport. *ACM Transactions on Graphics (Proc. SIGGRAPH 1997)* (1997), 65–76.

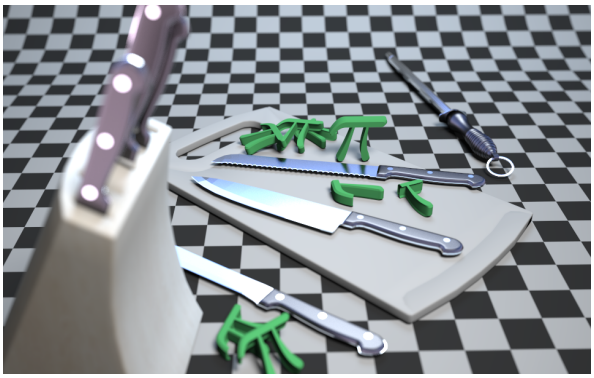
[WABG06] WALTER B., ARBREE A., BALA K., GREENBERG D.: Multidimensional lightcuts. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), pp. 1081–1088.

[War92] WARD G.: Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 265–272.

[WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D.: Lightcuts: a scalable approach to illumination. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (2005), pp. 1098–1107.

[WWZ*09] WANG R., WANG R., ZHOU K., PAN M., BAO H.: An efficient GPU-based approach for interactive global illumination. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers* (2009), pp. 1–8.

Figure 10: *Evolution of the error with increasing number of rendering passes compared to a converged image with* 11456 *rendering passes (resolution:* 1024×1024*). The top row shows the results after* $16, 128, 512$ *and* 2048 *rendering passes. The bottom row shows the inverse difference image scaled by* 8*. Figure 7 shows the graph of the errors for the full image series.*



Knives (960×600 1.7 hours): thin lens depth of field simulation

Interior (960×600 2.8 hours): many glossy materials and depth of field

Scene	Res.	Polygons	Avg. VPL	% Diffuse	% Eye Paths	% Caustic	Time per Pass
Box	1024×1024	49344	35.7	76.5%	5.7%	13.4%	4.8s
Museum	1200×520	331843	7.9	35.6%	21.7%	37.6.8%	1.2s
Interior	960×600	909441	3.8	43.4%	35.3%	9.3%	0.9s
Knives	960×600	95472	22.1	52.0%	6.2%	39.2%	3.1s

Figure 11: *The statistics show the relative time spent on the three techniques. The remaining time to* 100% *is due to management overhead. The projection map generation was between* 2% *and* 5% *of the total caustic time. The last column shows the average time per rendering pass (resolution dependent). It can be seen that in scenes with many glossy or singular materials the number of average VPL per pass drops while the time spent for the eye paths automatically increases.*

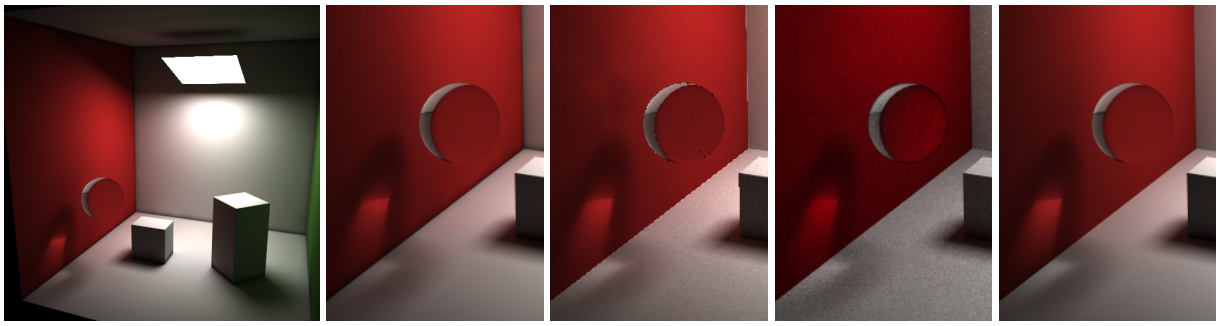


Figure 12: *Equal time comparison (*512×512*, 8 minutes) between our proposed algorithm (first), progressive photon-mapping (second) and metropolis light transport (third). The fourth image shows a pathtraced image with 300000 samples per pixel.*

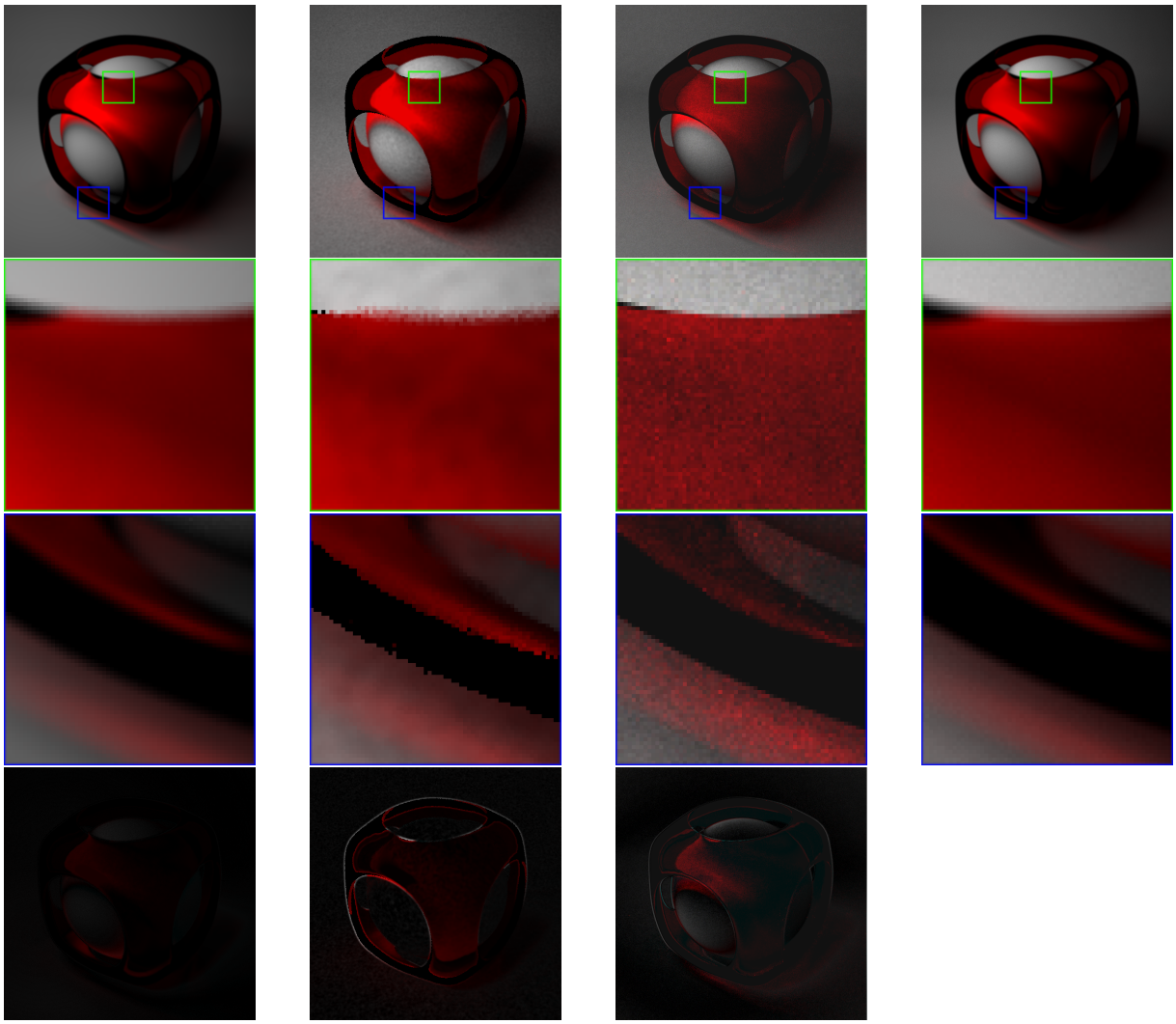


Figure 13: *Equal time comparison (*512×512*, 15 minutes) between our proposed algorithm (first), progressive photon-mapping (second) and metropolis light transport (third). The fourth image shows a pathtraced image with 300000 samples per pixel. The last row shows the difference to the path traced image.*