

Improving Image-Space Caustics Via Variable-Sized Splatting

*Chris Wyman**
University of Iowa

Carsten Dachsbacher
*University of Erlangen-
Nuremberg*

University of Iowa
Department of Computer Science
Technical Report UICS-06-02

*Primary Contact:
Department of Computer Science
14 MacLean Hall
The University of Iowa
Iowa City, IA 52242 USA

April 13, 2006

Abstract

Interactivity requires tradeoffs to achieve the right balance between rendering quality and speed. In practice, today's applications restrict lighting to mainly direct illumination, sometimes augmented by precomputed transfer techniques for diffuse global effects. Dynamic high-frequency specular effects, such as caustics, are largely lacking due to the high costs for recomputation each frame. Recent work has introduced a variety of related caustics approximations that interactively render light-space photons into a *photon buffer*, gather them into a *caustic map*, and project this map onto the scene similar to shadow mapping. While the process is simple and straightforward, the discretization of light into a finite number of uniformly-distributed photons leads to undersampling and aliasing artifacts. This paper examines two techniques for reducing these artifacts using varying sized photon splats. Conceptually, these are similar to the variable-radius k -nearest neighbor search used in photon mapping, allowing noise reduction in areas of low photon density while maintaining crisp caustics at focal points. Our techniques improve image quality at a modest cost that is significantly cheaper than supersampling the photon buffer.

1 Introduction

Image synthesis techniques using global illumination seamlessly render many effects, such as soft shadows, diffuse interreflections, and caustics, that prove difficult to add to local illumination models. Unfortunately, due to finite computation budgets, speed typically trumps realism in interactive applications. Because global illumination requires significant computational resources, only precomputed techniques and methods for specific effects (e.g., shadows) have found frequent use in mainstream interactive applications.

Computer graphics researchers have long attempted to quickly render realistic images, so the literature is rich with methods for speeding up standard radiosity [9], Monte Carlo path tracing [15], and photon mapping [14] techniques for global illumination. In recent years, all of these approaches have been adapted to take advantage of the nearly-ubiquitous graphics accelerators in modern PCs. However, these accelerated implementations still run too slowly for widespread adoption.

While incremental radiosity techniques [4] avoid explicitly storing and solving large linear systems, GPU-based algorithms [3, 5] must invert the standard progressive refinement “scatter” operation into a more GPU-friendly “gather” operation. This limits the scalability as scene complexity increases, as gathers must occur at each scene element. Additionally these GPU techniques continue to limit the use of non-Lambertian surfaces.

Path and ray based approaches typically rely on recursive tracing of paths through a scene. While this is an elegant concept that seamlessly incorporates complex primitives, lights, and materials, it does not port well to the GPU’s peculiar stream processing model. GPU-based ray tracing [18] and path tracing [2] is possible, but inherent ray incoherency limits rendering speeds on hardware designed for incremental rasterization.

Photon mapping is a two pass process, where the first pass shoots photons from light sources and the second pass renders from the eye’s view, gathering photons near each visible point. This two-pass approach fits well with standard GPU-based multipass rendering techniques, as both passes can be framed as coherent visibility queries from a single viewpoint and the second pass already uses gather instead of scatter operations. While implementations of complete GPU-based photon mappers exist [17], they rely on incoherent ray based techniques for secondary bounces and complex data structures for photon storage, limiting the overall speed.

This paper improves on recent work [21, 20, 22, 27, 30] inspired by photon and shadow mapping. In these approaches the scene is rendered twice, once from each the light and the eye. In both rendering passes only specular bounces are considered, borrowing from recent advances in GPU-based specular rendering [22, 29]. This eliminates the highly incoherent secondary diffuse rays captured by Purcell et al. [17] at the loss of diffuse interreflections in the final image. Instead of storing photons in a complex grid or KD-tree structure, photons are stored in a simple 2D texture—the framebuffer. These two simplifications allow interactive rendering of reflective and refractive caustics.

However, these *caustic mapping* techniques rely on limited types of photon distributions. Shah and Pattanaik [21] only shoot photons towards the vertices of reflectors and refractors. The other techniques use a regular sampling from the light’s point of view, where many photons completely miss the specular geometry (and are useless for caustic computations). Either approach introduces undersampling and aliasing artifacts, leading to noise in the result. While Gaussian splats and frame-to-frame filtering reduce noise in the spatial and temporal domains, the results still tend to be either noisy [30] or very blurry [22].

Ultimately, the problem is that previous techniques do not perform a complete k -nearest neighbor search, but rather depend on fixed-width splats or filter kernels. Since searching for nearest neighbors in the full-resolution photon buffer is expensive, this paper instead suggests approximating the variable-radius k -nearest neighbor search using either adaptive multi-resolution splatting or varying splat sizes dynamically based on refractor properties. As Figure 1 shows, significant noise reduction in undersampled regions is possible while maintaining caustic crispness near focal points.

The rest of the paper is organized as follows. Section 2 discusses previous interactive caustics techniques and Section 3 briefly describes image-space caustic methods. Section 4 discusses image-space noise reduction and introduces our proposed improvements. Results are presented in Section 5, followed by a few concluding remarks.

2 Previous Work

Techniques for realistically rendering caustics have been around for at least twenty years, but only recently has computational power improved to the point where reasonable approximations run interactively. Typically, these methods rely either on

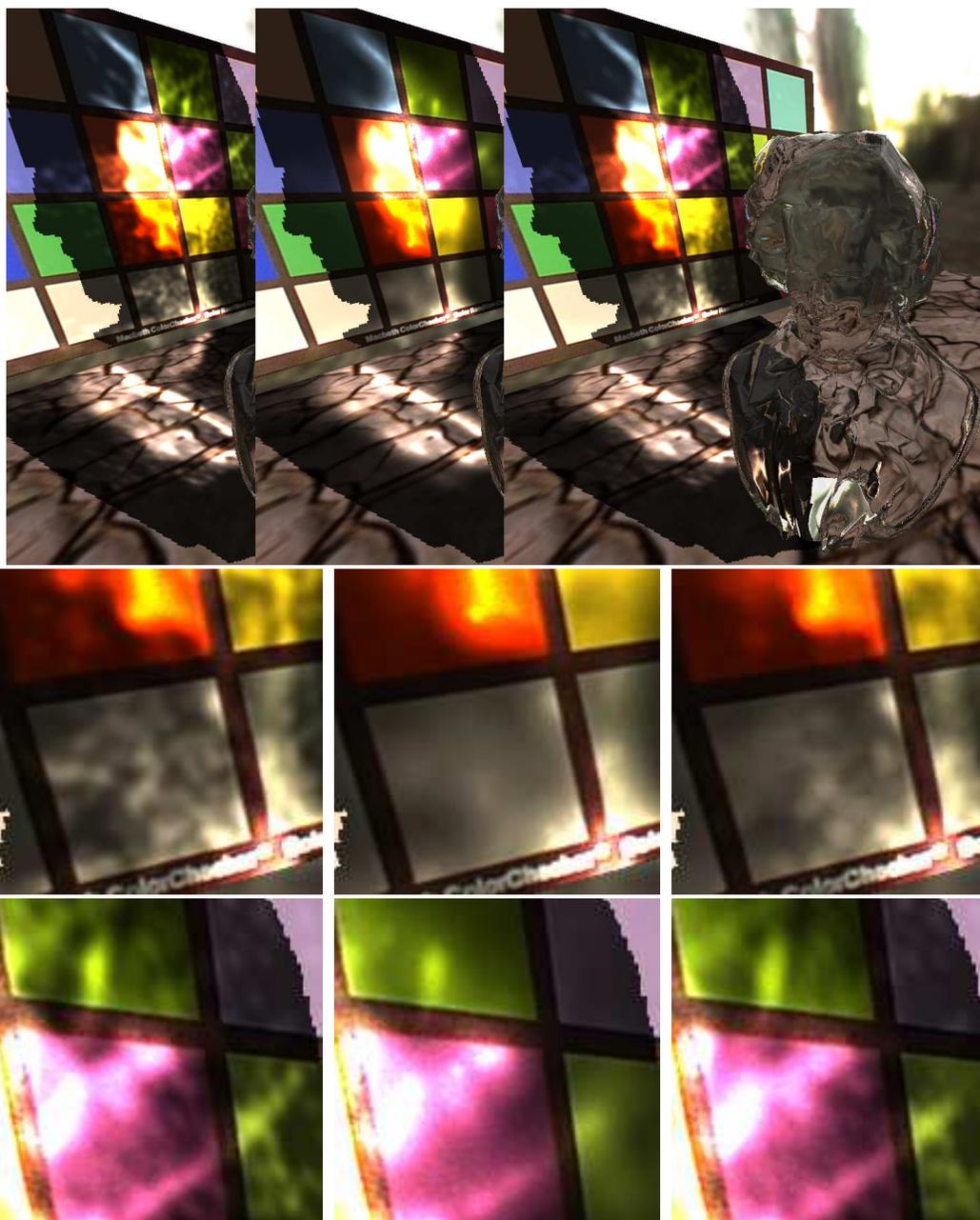


Figure 1: Compare caustics rendered with (left) the approach of Wyman and Davis [30], and the approaches discussed in (center) Section 4.1 and (right) Section 4.2. The bottom rows show closeups of two regions.

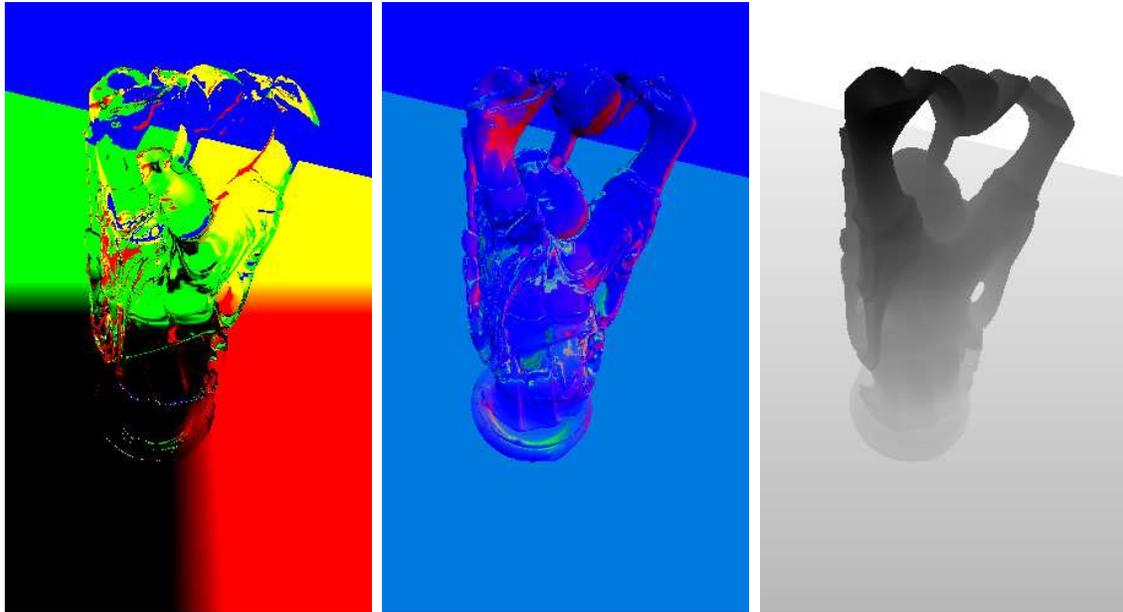


Figure 2: *Buffers created during the first pass, rendering from the light's point of view: (left) final photon positions, (center) the photon's incident direction at its final position, and (right) a standard depth map for rendering shadows.*

particle tracing [28, 15] or beam tracing [11] to compute caustic intensity.

Using beam tracing to generate caustics was introduced by Watt [26], who shot beams backwards from the light. Each beam was defined by a light source and three triangle vertices on a tessellated water surface, was reflected or refracted by the surface, and intersected with a diffuse receiver. The caustic polygons, defined by beam intersections with diffuse surfaces, were later blended with a standard diffuse rendering. Nishita and Nakamae [16] accounted for light-water scattering throughout these refracted prismatic beams and Iwasaki et al. [12, 13] examined ways to accelerate rendering using graphics hardware. Ernst et al. [8] improved these results by rendering warped volumes instead of prisms. Unfortunately, all these approaches limit caustic beams to a single specular bounce.

The particle tracing techniques stem from backwards ray tracing [1] and photon mapping [14], which both emit photons from light sources and store their intersections with geometry in either a 2D or 3D structure. Current interactive particle tracing techniques either reuse old cached data until it can be updated [23], incrementally shoot a budgeted number of photons each frame [24], regenerate important photon paths each frame [10] using parallel CPU clusters and selective photon tracing [7], or

precompute a complex caustic volume [31] to index into dynamically. Unfortunately, all these techniques have sizable memory requirements, sizable CPU-based computation budgets, or both. Wand and Straßer [25] proposed a slightly different scheme, which finely tessellates reflectors and performs a per-pixel loop to sum contributions of all reflector polygons. While this approach allows complex, area lights it scales linearly with increased reflector tessellation, limiting geometric complexity.

3 Image-Space Caustic Overview

Recent flexibility in programmable graphics hardware has allowed improvements in interactive particle-based caustics [21, 20, 22, 27, 30]. These techniques render the scene from the light’s viewpoint, storing a standard depth map, final locations of reflected or refracted photons, and each photon’s incident direction at its final position (see Figure 2). The photons are redrawn into a light-space *caustic map* that can be projected onto the scene similar to a shadow map [21, 20, 30], as in Figure 3. An alternative approach [22, 30] gathers photons directly in the final eye-space rendering. We do not consider this approach in the rest of the paper, but our improvements could be adapted in a straightforward manner.

Interestingly, these methods all use different numbers of photons and techniques for emission from the light. Shah and Pattanaik [21] traced one photon through each vertex on a refractor, Szirmay-Kalos et al. [22] and Shah et al. [20] used photons from a uniform grid of size 32^2 or 64^2 , whereas Wyman and Davis [30] focused on larger 512^2 to 2048^2 uniform grids.

Obviously, each choice has its advantages and disadvantages. Tracing one photon per vertex avoids uniform sampling artifacts seen with the other approaches, but fails to generate accurate results in coarsely tessellated models. Using a uniform grid with a small number of photons adds relatively little overhead but requires significant smoothing (e.g., with large Gaussian splats), eliminating sharpness in regions where many photons converge. Shooting sizable numbers of photons increases computational overhead, but allows a smaller splat that maintains crispness near focal points at the expense of increased noise in undersampled regions.

This paper extends the work of Wyman and Davis [30] by varying the size of the splat to help reduce noise. Our approach uses small splats near sharp focal points to maintain crispness, while reducing noise in sparsely sample regions by drawing larger splats.

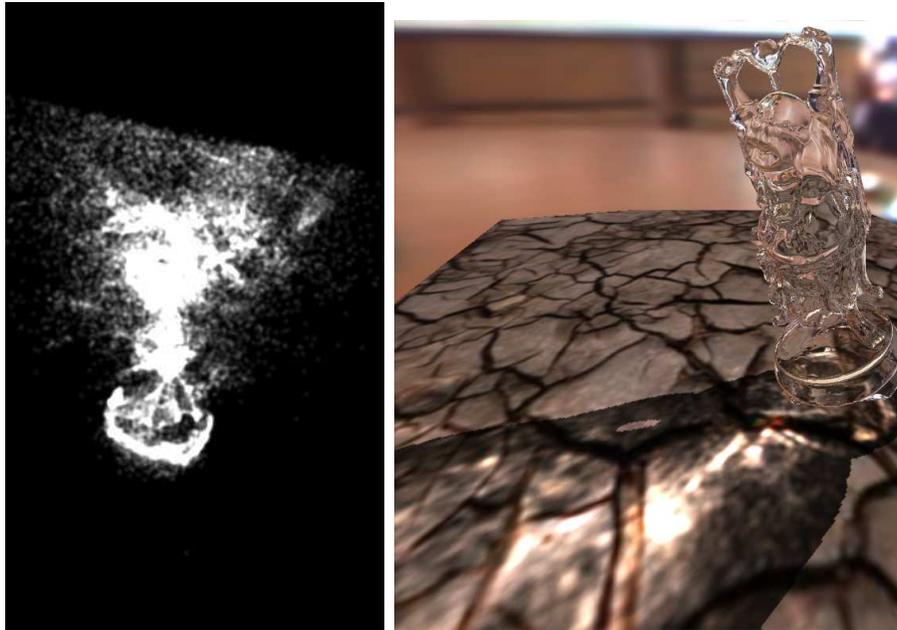


Figure 3: *Caustics rendered in light space. Photons are drawn into the caustic map and blurred (right) and are then projected onto the scene similar to shadow mapping (left).*

4 Noise Reduction in Image-Space

Before discussing our techniques for dynamically varying the photon search radius, it is instructive to examine a standard approach for noise reduction in image-based algorithms: mipmapping. After all, mipmapping provides a cheap and simple technique for adaptively blurring images. One obvious approach would dynamically generate a mipmap for the caustic map, and index into the pyramid based on photon density instead of distance to the eye.

Unfortunately, this method exhibits a few problems. First, mipmapping applies successive box filters to the initial caustic map, but such averaging is not the most efficient filter to reduce noise from photon mapping [14, 19]. Second, close proximity does not guarantee that nearby photons will be averaged together at a low mipmap level, due to the mipmap generation process. For instance, photon contributions will not bleed across the center of image until the highest level, which leads to artifacts and popping when combining contributions from multiple levels.

One way to avoid this problem would splat photons into each mipmap level, as shown in Figure 4, instead of using hardware mipmap generation. This approach eliminates the mipmapping problems discussed above, but increases the cost of rendering an M^2 caustic map with N^2 photons from $O(N^2)$ to $O(N^2 \log M)$, since each photon must be rendered into each mipmap level.

Examining Figure 4 shows that if the expense could be mitigated, there is some merit to this idea. At the lowest mipmap level, where 512^2 photons are splatted into a 512^2 buffer, there is a lot of noise. However, if the caustic contribution is scaled by a factor of $\frac{1}{16}$, very little noise remains. This makes sense—regions where photons congregate should be filtered with a relatively small filter. However, increasing this filter by splatting the same photons into coarser images gives better results in less densely populated regions while over blurring focal regions.

Section 4.1 describes an approach based off these mipmap observations, whereas Section 4.2 introduces a technique that allows splat sizes to vary continuously, rather than in discrete steps.

4.1 Adaptive Multi-Resolution Splatting

Examining the images in Figure 4 reveals that noise is virtually eliminated where the number of photons emitted exceeds the caustic map resolution by a factor of 16 (e.g., 512^2 photons splatted into a 128^2 image). Also, the noise disappears once roughly 16 photons land in each texel of the caustic map. This suggests simply generating a subset of the mipmap pyramid, using the finest resolution map in regions where photons focus and using progressively coarser resolution maps as the photon density drops.

To avoid splatting each photon once into each mipmap level, we propose the following. This approach replaces the mipmap with two full-resolution caustic maps, but only splats each photon a single time:

1. Render each photon as a single, large point of radius R .
2. In the fragment shader, compute the intensity for Gaussian splats of four different radii r_i , where $r_i \leq R$.
3. Render into the two caustic maps using multiple render targets. Both should have additive blending enabled.

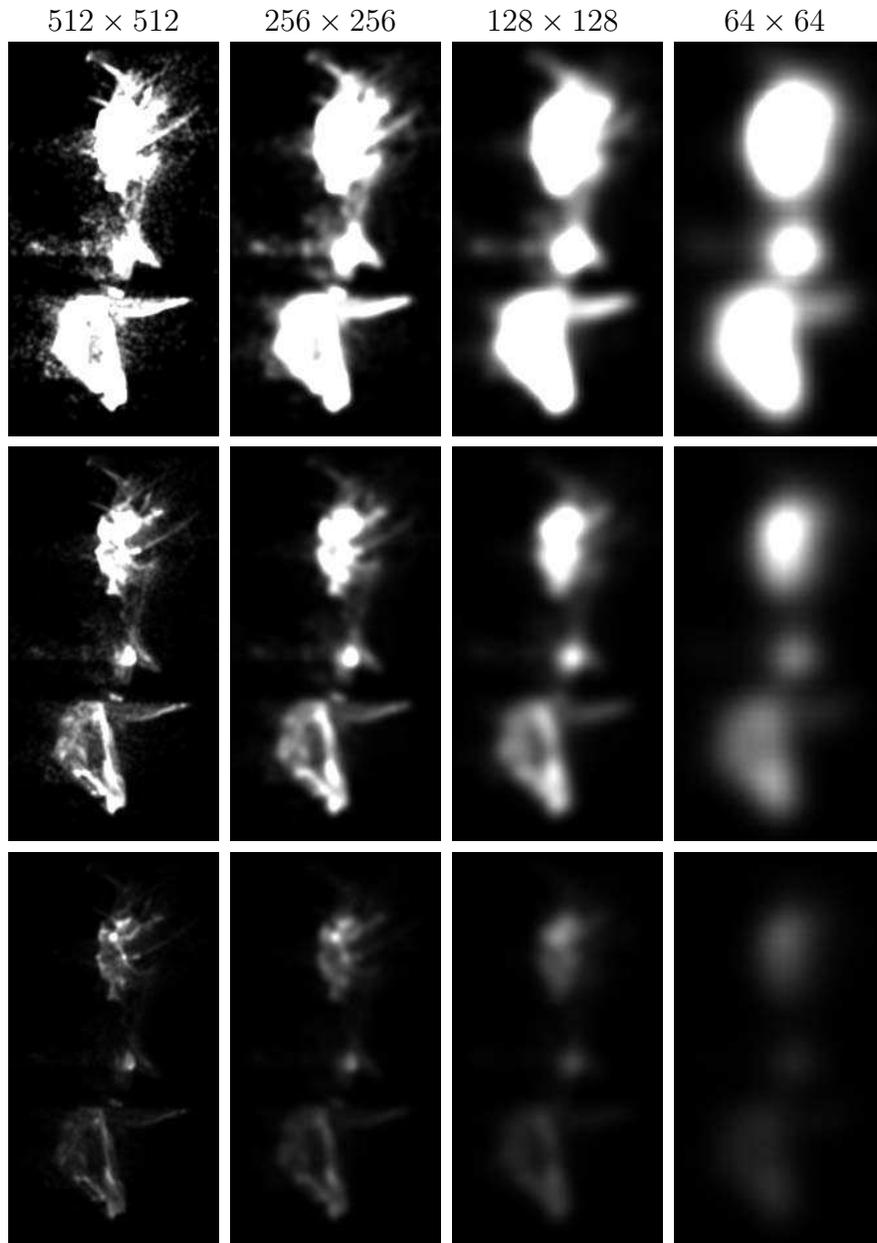


Figure 4: *Top*: 512^2 photons accumulated into caustic maps of varying resolution. *Center*: The same images with caustic contributions scaled by $\frac{1}{4}$. *Bottom*: The top images, scaled by $\frac{1}{16}$. These images show what one would expect: focal regions should use small filters to maintain sharp caustics, but sparsely sampled regions should use large filters to reduce noise.

- (a) In the first buffer, output the four intensities of the photon splats in the four separate channels.
 - (b) In the second buffer, for each channel output a 1 if the texel was inside the corresponding photon splat and a 0 if the texel was outside the splat.
4. When projecting the caustic map onto the scene, first examine the photon density from the second buffer, and pick the splat from the first buffer based on this count.

One advantage to this approach is that filter sizes are no longer dependent on mipmap image resolution. Our implementation uses Gaussians roughly corresponding to OpenGL point sizes of 3, 7, 11, and 15. Wyman and Davis [30] used a 7×7 Gaussian filter, so our approach allows both finer detail in focal regions and additional smoothing in sparsely sampled areas.

A disadvantage of this approach is that it requires larger point sizes (e.g., 15^2 instead of 7^2). Due to the number of photons used, however, the original approach is not fragment bound. This allows the increased splat size without a reduction in performance (though beyond a 15^2 point size, our fragment shader became the bottleneck).

In order to avoid sharp discontinuities when switching between splats of different radii, some interpolation is necessary. Unfortunately, a straightforward linear interpolation leads to ringing artifacts, as shown in Figure 5. When photon density drops, the intensity given by smaller filters drops quicker than that of the larger filters. This means caustic intensity is not guaranteed to strictly increase in the interpolation region. Interpolating only when the smaller splat has a higher intensity improves results significantly. Furthermore, we found density dropped quickly enough in some regions that we needed to consider contributions from three sized splats. For this reason, we avoid a linear weighting and used a Bernstein interpolant that increases as splat photon counts increase from a minimum threshold (8 photons) to a maximum (16 photons), normalized based on the number of splats weighted.

Conceptually, this method uses a variable-sized neighborhood search with four preset search radii, instead of the fixed search radius of previous work. While this dramatically reduces noise, it restricts splat sizes to those chosen *a priori*, requires a physically inaccurate interpolation scheme, and requires a second gather pass that interpolates between the discrete splats. The next section introduces a different approach that allows gathering in a single pass, with continuously varying splats based on material properties.

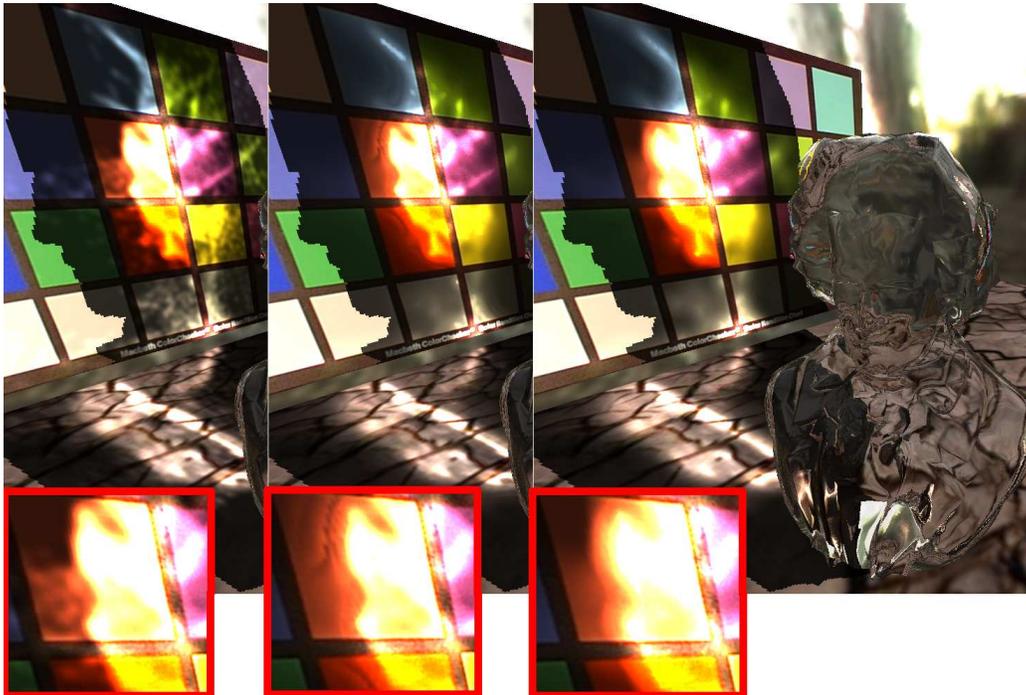


Figure 5: Compare (left) the previous caustics approach with (center) the multi-resolution splat approach with linear interpolation between splats and (right) interpolation using the Bernstein interpolant only in regions where smaller splats increase the caustic intensity.

4.2 Using Material Properties to Vary Splat Sizes

The gather pass described above draws inspiration from the nearest k -neighbor search in photon mapping, where photon density determines the search radius. An alternative approach instead relies on beam tracing techniques [11], where a photon's region of influence depends on how it is warped by the refractor along the path from light to receiver. By considering appropriate material properties, we can determine the correct region of the receiver affected by the photon, allowing each photon to independently choose a correct splat size. This is inspired by recent work by Dachsbacher and Stamminger [6], who used a smaller number of dynamically sized splats to achieve interactive diffuse illumination and caustics from single-bounce reflections.

In the case of a refractive caustic, first consider the standard elementary optics problem shown in Figure 6. Light emitted towards the idealized thin lens travels a distance d_{front} to the lens, refocuses at some point (in this case, behind the lens) a distance

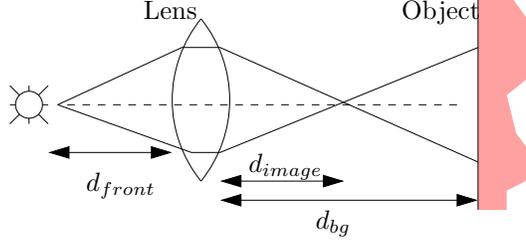


Figure 6: A spherical thin lens focuses light emitted from a point source at a distance d_{front} from the lens back to a point a distance d_{image} from the lens, where d_{front} and d_{image} are related by Eq. 1. For caustics, the important information is the beam size when it hits a receiver object at distance d_{bg} .

d_{image} from the lens, and hits a background object at distance d_{bg} . The distances d_{front} and d_{image} are related using the thin lens equation:

$$\frac{1}{d_{front}} + \frac{1}{d_{image}} = \frac{1}{f}, \quad (1)$$

where f is the lens' focal length, and is given by the lens maker's equation:

$$\frac{1}{f} = (n_{in} - n_{out}) \left(\frac{1}{r_{front}} - \frac{1}{r_{back}} \right). \quad (2)$$

Here n_{in} and n_{out} are the indices of refraction inside and outside the lens and r_{front} and r_{back} are the radii of curvature for the front and back faces of the lens. Once the image distance d_{image} is known, the lens' magnification M is given by:

$$M = -\frac{d_{image}}{d_{front}} = \frac{f}{f - d_{front}} = \frac{f - d_{image}}{f}. \quad (3)$$

We are interested in the magnification of the beam because regions of high magnification receive more light so photon splats should be smaller, conversely areas of minification receive less light and require larger splats. Unfortunately, the magnification given by Equation 3 refers to the magnification at distance d_{image} from the lens, when the light source is in focus. Instead, we need to determine the size of the photon's "beam" when it hits the surface a distance d_{bg} from the lens. Consider the situation in Figure 7. The area of the photon beam that hits the lens (A_{lens}) is $d\omega_{ph}d_{front}^2$, the area of the beam that hits the background object (A_{bg}) is $d\omega_{refr}(d_{bg} - d_{image})^2$, and the area of an unrefracted beam that hits the background (A_{orig}) is $d\omega_{ph}(d_{front} + d_{bg})^2$, assuming a thin lens. So the beam magnification M_{beam} at the background object is

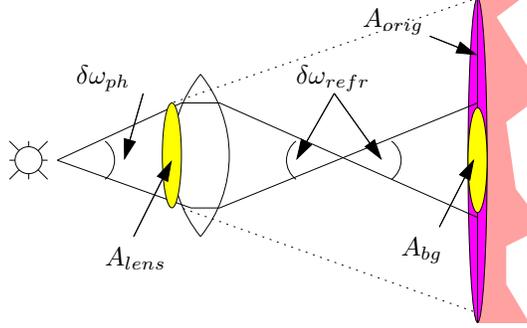


Figure 7: Given a photon subtending a solid angle of $d\omega_{ph}$, it hits a region of the lens of area A_{lens} . In a thin lens, the area exiting the lens is the same, but now subtends a solid angle of $d\omega_{refr}$ from the image point. The refocused beam intersects the receiver at distance d_{bg} with some area A_{bg} instead of the unrefracted area A_{orig} .

given by:

$$M_{beam} = \frac{A_{bg}}{A_{orig}} = \frac{d\omega_{refr}(d_{bg} - d_{image})^2}{d\omega_{lens}(d_{front} + d_{bg})^2}. \quad (4)$$

Using the thin lens approximation, the beam's area remains the same between the entrance and exit of the lens, giving

$$d\omega_{ph}d_{front}^2 = A_{lens} = d\omega_{refr}d_{image}^2. \quad (5)$$

Replacing $d\omega_{refr}$ in Equation 4 gives

$$M_{beam} = \frac{d_{front}^2}{d_{image}^2} \frac{(d_{bg} - d_{image})^2}{(d_{front} + d_{bg})^2}, \quad (6)$$

but because OpenGL requires us to specify the point diameter rather than its area, we can take the square root, giving diameter D_{new} :

$$D_{new} = \left(\frac{d_{front}}{d_{front} + d_{bg}} \frac{d_{bg} - d_{image}}{d_{image}} \right) D_0. \quad (7)$$

Here D_0 is the default diameter of non-refracted photon splats, which depends on the ratio between the photon buffer and caustic map resolutions. Arbitrarily increasing D_0 decreases caustic noise at the cost of increased blur.

For each photon, given the distances d_{front} and d_{bg} , the indices of refraction n_{in} and n_{out} , the radii of curvature r_{front} and r_{back} , the unrefracted splat size D_0 , and Equations 1, 2, and 7 a splat size for the refracted photon can be determined. This can be rendered as described previously [30], but using this modified splat area.

As presented, there are two major assumptions in this approach: we approximate both front and back refractors as ideal spherical refractors, and we treat the refractor as a thin lens. The second assumption can easily be eliminated using the thick lens equations, which replace Equations 1 and 2 with slightly more complex computations.

5 Results and Discussion

We implemented our noise reduction techniques in OpenGL using Cg `vp40` vertex and `fp40` fragment shaders. The prototype code is relatively unoptimized, with the implementation designed for flexibility and display of intermediate results, rather than speed. Timings were taken on a dual-core 3.0 GHz Pentium 4 processor with a 256 MB nVidia GeForce 7800 GTX.

Because we splat photons directly into the caustic map, instead of performing an image-space blur as proposed by Wyman and Davis [30], our baseline timings (see Table 1) are roughly 40% higher than their reported speeds. Even with our noise reduction techniques, all of our timings are faster than the numbers they reported. However, our improvements do impose an overhead ranging between 15 and 40%. Typically, continuously varying the splat size results in better performance than interpolation between fixed sized splats, as contributions for the multiple splats must be temporarily stored for an additional interpolation pass. In fragment bound examples, further speedups occur as many splats cover only a few pixels.

Figures 1 and 8 show comparisons between the previous method and our noise reduction schemes. For static images, this noise reduction is most pronounced on very simple and very complex geometry. On simple geometry such as a sphere, discretization errors in the refraction can cause ringing artifacts. These artifacts are nearly completely eliminated using both techniques. For very complex geometry, 512^2 photons does not sufficiently sample the lighting, leading to severe undersampling noise. In these cases our methods partially but dramatically reduce the noise.

Generally, the multi-resolution splat approach over blurs the caustic slightly while the continuously varying approach leaves more, but relatively uniformly distributed, noise. We found during animation the uniformly distributed noise from the continuous scheme is less objectionable than the noise from the multi-resolution approach, where interpolation regions can cause odd intensity gradients.

Note that both approaches are orthogonal to supersampling and the temporal filtering

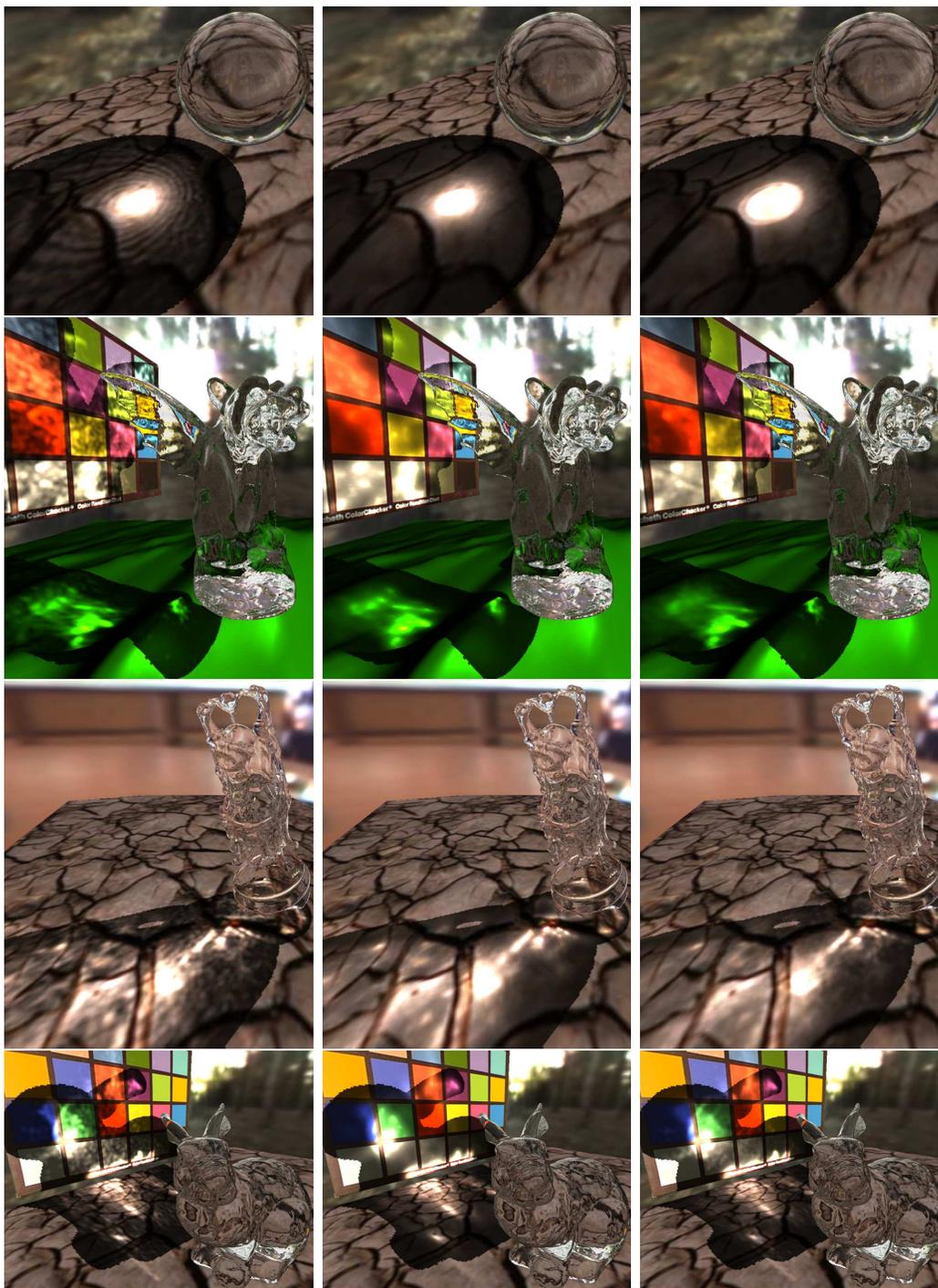


Figure 8: Various scenes rendered with (left) the approach of Wyman and Davis [30], (center) the multiple resolution splat technique, and (right) determining splat sizes dynamically using the lens equations. All images use 512^2 photons.

Scene	# Photons	Prev. Work	Multi Res	Thin Lens
Ball & Dragon (256K triangles)	512 ²	25.1	21.5	19.5
	1024 ²	11.3	9.5	8.0
Beethoven (5K triangles)	512 ²	49.4	27.7	38.8
	1024 ²	13.4	7.9	11.1
Buddha (50K triangles)	512 ²	35.1	25.6	26.8
	1024 ²	13.5	8.6	8.5
Bunny (70K triangles)	512 ²	30.8	26.4	27.6
	1024 ²	12.9	9.6	10.3
Gargoyle (205K triangles)	512 ²	16.3	13.3	14.1
	1024 ²	8.4	6.4	6.2

Table 1: *Timings for scenes from the paper and accompanying video. Values are frames per second for a 512² image resolution, with either 512² or 1024² photons emitted from the light. Benchmarks used dynamic scenes, where the photon buffer and caustic map were recomputed every frame.*

suggested by Wyman and Davis. The accompanying video shows that combining temporal filtering with either of our proposed approaches reduces noise to virtually undetectable levels and maintains crispness using only 512² photons per frame. Note, however, that all images in the paper use photons from a single frame to better highlight the improvement.

6 Conclusion

This paper presented two approaches to reducing noise in existing image-space caustics techniques. These ideas derive inspiration from standard photon mapping, where photons are gathered in variable sized regions around every point. However, our variable-sized gathers can be easily implemented using the feed-forward pipeline of standard graphics accelerators. While we implemented our approach using a framework similar to Wyman and Davis [30], our methods could easily adapted to work with the techniques of Szirmay-Kalos et al. [22], Wei and Kaihuai [27] or Shah et al. [21, 20].

In some ways, our work can be seen as a convergence of beam-tracing caustic techniques and particle-based schemes. However, in this regard there are some limitations.

In particular, because we use a spherical lens approximation, we only consider a single curvature value at each pixel. Real surfaces often have two different principal curvatures that must be accounted for. Furthermore, despite varying the splat size using a magnification metric, high quality renderings still require sampling sufficient numbers of photons.

Finally, we inherit a number of limitations from the underlying techniques. Our refraction is limited to two interfaces, without total internal reflection. Our ray-background intersection approximation is relatively slow to get high quality intersections necessary for sharp caustics. And our photon buffer contains numerous unrefracted photons discarded by the gather process (see Figure 2), but which must still be processed by the vertex shader. Future work easing these limitations should vastly improve rendering speed and quality.

References

- [1] ARVO, J. Backward ray tracing. *Developments in Ray Tracing* (1986), 259–263. ACM Siggraph '86 Course Notes.
- [2] BARSÍ, A., SZIRMAY-KALOS, L., AND SZIJARTO, G. Stochastic glossy global illumination on the gpu. In *Proceedings of the 21st spring conference on computer graphics* (2005), pp. 187–193.
- [3] CARR, N., HALL, J., AND HART, J. Gpu algorithms for radiosity and subsurface scattering. In *Proceedings of Graphics Hardware* (2003), pp. 51–59.
- [4] CHEN, S. E. Incremental radiosity: An extension of progressive radiosity to an interactive image synthesis system. In *Proceedings of SIGGRAPH* (1990), pp. 135–144.
- [5] COOMBE, G., HARRIS, M. J., AND LASTRA, A. Radiosity on graphics hardware. In *Proceedings of the Graphics Interface* (2004), pp. 161–168.
- [6] DACHSBACHER, C., AND STAMMINGER, M. Splatting indirect illumination. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2006), pp. 93–100.
- [7] DMITRIEV, K., BRABEC, S., MYSZKOWSKI, K., AND SEIDEL, H.-P. Interactive global illumination using selective photon tracing. In *Proceedings of the Eurographics Workshop on Rendering* (2002), pp. 25–36.
- [8] ERNST, M., AKENINE-MÖLLER, T., AND JENSEN, H. W. Interactive rendering of caustics using interpolated warped volumes. In *Proceedings of Graphics Interface* (2005), pp. 87–96.
- [9] GORAL, C., TORRANCE, K., GREENBERG, D., AND BATTAILE, B. Modelling the interaction of light between diffuse surfaces. In *Proceedings of SIGGRAPH* (1984), pp. 213–222.
- [10] GÜNTHER, J., WALD, I., AND SLUSALLEK, P. Realtime caustics using distributed photon mapping. In *Proceedings of the Eurographics Symposium on Rendering* (2004), pp. 111–121.

- [11] HECKBERT, P. S., AND HANRAHAN, P. Beam tracing polygonal objects. In *Proceedings of SIGGRAPH* (1984), pp. 119–127.
- [12] IWASAKI, K., DOBASHI, Y., AND NISHITA, T. An efficient method for rendering underwater optical effects using graphics hardware. *Computer Graphics Forum* 21, 4 (November 2002), 701–711.
- [13] IWASAKI, K., DOBASHI, Y., AND NISHITA, T. A fast rendering method for refractive and reflective caustics due to water surfaces. *Computer Graphics Forum* 22, 3 (September 2003), 601–609.
- [14] JENSEN, H. W. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.
- [15] KAJIYA, J. The rendering equation. In *Proceedings of SIGGRAPH* (1986), pp. 143–150.
- [16] NISHITA, T., AND NAKAMAE, E. Method of displaying optical effects within water using accumulation buffer. In *Proceedings of SIGGRAPH* (1994), pp. 373–381.
- [17] PURCELL, T., DONNER, C., CAMMARANO, M., JENSEN, H. W., AND HANRAHAN, P. Photon mapping on programmable graphics hardware. In *Proceedings of Graphics Hardware* (2003), pp. 41–50.
- [18] PURCELL, T. J., BUCK, I., MARK, W. R., AND HANRAHAN, P. Ray tracing on programmable graphics hardware. *ACM Transactions of Graphics* 21, 4 (2002), 703–712.
- [19] RUSS, J. C. *The Image Processing Handbook*, fourth edition ed. CRC Oress, Boca Raton, Florida, 2002.
- [20] SHAH, M., KONTTINEN, J., AND PATTANAIK, S. Caustics mapping: An image-space technique for real-time caustics. *IEEE Transactions on Visualization and Computer Graphics*, (to appear).
- [21] SHAH, M., AND PATTANAIK, S. Caustics mapping: An image-space technique for real-time caustics. Tech. Rep. CS-TR-50-07, University of Central Florida, August 2005.
- [22] SZIRMAY-KALOS, L., ASZODI, B., LAZANYI, I., AND PREMECZ, M. Approximate ray-tracing on the gpu with distance impostors. *Computer Graphics Forum* 24, 3 (2005), 685–704.
- [23] TOLE, P., PELLACINI, F., WALTER, B., AND GREENBERG, D. Interactive global illumination in dynamic scenes. In *Proceedings of SIGGRAPH* (2002), pp. 537–546.
- [24] WALD, I., KOLLIG, T., BENTHIN, C., KELLER, A., AND SLUSALLEK, P. Interactive global illumination using fast ray tracing. In *Proceedings of the Eurographics Rendering Workshop* (2002), pp. 15–24.
- [25] WAND, M., AND STRASSER, W. Real-time caustics. *Computer Graphics Forum* 22, 3 (2003), 611–620.
- [26] WATT, M. Light-water interaction using backward beam tracing. In *Proceedings of SIGGRAPH* (1990), pp. 377–385.
- [27] WEI, H., AND KAIHUAI, Q. Interactive approximate rendering. Tech. Rep. CS TR 0815-05, Department of Computer Science and Technology, Tsinghua University, China, 2005.

- [28] WHITTED, T. An improved illumination model for shaded display. *Communications of the ACM* 23, 6 (June 1980), 343–349.
- [29] WYMAN, C. An approximate image-space approach for interactive refraction. *ACM Transactions on Graphics* 24, 3 (July 2005), 1050–1053.
- [30] WYMAN, C., AND DAVIS, S. Interactive image-space techniques for approximating caustics. In *Proceedings of ACM Symposium on Interactive 3D Graphics and Games. (to appear)* (March 2006).
- [31] WYMAN, C., HANSEN, C., AND SHIRLEY, P. Interactive caustics using local precomputed irradiance. In *Proceedings of the Pacific Conference on Computer Graphics and Applications* (October 2004), pp. 143–151.