Rendering Smooth Spectrum Caustics on Plane for Refractive Polyhedrons

Peter Sikachev, Ilya Tisevich, Alexey Ignatenko
Department of Computational Mathematics and Cybernetics
Moscow State University, Moscow, Russia
{psikachev, itisevich, ignatenko}@graphics.cs.msu.ru

Abstract

Caustics are patterns of light formed by refraction or reflection of light from objects, and several methods have been developed to render this effect. Some methods are used to render caustics on plane, others – on arbitrary surface.

Practically no methods have been developed to render spectrum caustics, which appear when refracting material has too high refraction index or ray suffers multiple refractions and reflections before it exits the object. Existing methods, like photon mapping [11] either produce non-smooth results for few wavelengths number or work offline.

We propose a method for rendering spectrum caustics on plane. The caustics are calculated using forward vertex ray-tracing algorithm and rendered using conventional scanline rasterization (OpenGL). Besides, we propose a technique that allows rendering smooth caustics, even when only 3 rays per vertex are traced, of wavelengths roughly corresponding to conventional color components (R, G, B).

Keywords: *Caustics, Ray-Tracing, Spectrum Rendering.*

1. INTRODUCTION

Rendering caustics is a tricky problem because of multiple factors. First, if an object is refractive (and *diacaustic* is supposed), one or more refractions should be simulated, which could be time-consuming.

Then, there is a problem of projecting caustics on the desired surface. This can become difficult if the surface is non-planar.

Finally, the problem of rendering caustics arises. Even in case of per-pixel ray-casting, the surface would be irregularly lit (due to curvature of the caustic producing object) and methods of interpolation between single photons are required.

In our work we limit ourselves to the case of polyhedron object and a planar surface, on which caustics are projected. Our model doesn't differentiate between reflection and refraction cases, that is why it is possible to render both catacaustics and diacaustics using it.

Our contributions are an algorithm for spectrum caustics rendering and an algorithm for rendering smooth spectrum caustics tracing only few rays.

2. RELATED WORK

Caustic effect now is widely used in high-quality computer graphics, be it interactive [1-4, 7-10] or not [6]. There is also a difference in assumptions that are made to obtain desired quality.

While rendering diacaustics, the method of refraction calculation is vital. The fastest methods only take one refraction into account [2, 4]. [1] introduces a tricky technique which allows approximate rendering of up to two refractions without using ray-tracing or ray-casting. Techniques based on photon mapping or ray-depth map intersection algorithms [8] can handle multiple reflections and refractions within the object.

Different assumptions about caustics receiver are made. Some works [2] assume it to be a plane. Other techniques render caustics on plane and then use it as a projective texture [1]. Finally, two-plane parameterization (2PP) is used to render caustics most precisely [8].

Practically no methods have been proposed recently for interactive rendering of spectrum caustics. Usually, in case of few reflections/refractions, only white caustics appear, but this was not our case.

We have developed a method for rendering spectrum caustics, casted from a polyhedron object onto a plane. We use a forward vertex ray-tracing algorithm to trace a ray through a reflective object with high accuracy, which is described in Section 3.

Caustic rendering algorithm itself is shown in Section 4.

High-quality version of our algorithm produces smooth caustics even from three rays. It is discussed in Section 5.

We conclude and summarize our plans for nearest future research in Section 7.

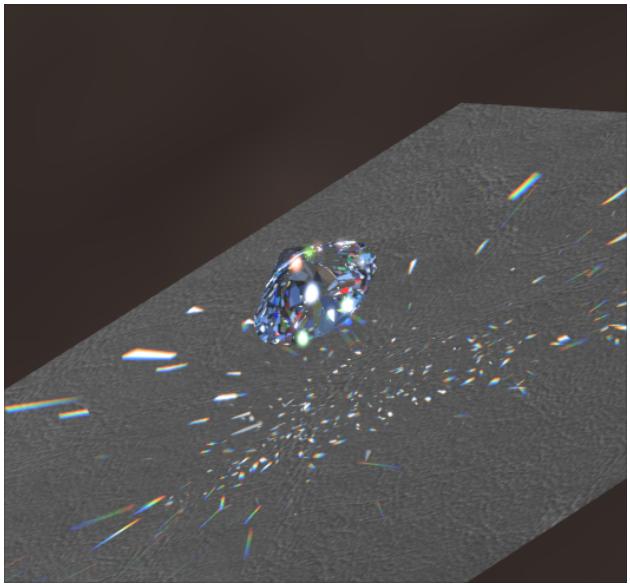

Figure 1: Example of caustics on a plane for 3 rays for typical object with high refraction index.

3. FORWARD VERTEX RAY-TRACING

The ray tracing algorithm used is a conventional physically-correct polyhedron ray tracer. Actual Fresnel formulas are used to

calculate ray reflection and refraction at every ray-polygon hit. Each single ray has a specific wavelength and intensity assigned to it. Provided with a set of physical properties of the polyhedron material, the algorithm can accurately calculate light distribution within the object, and provide us with directions and intensities of exiting rays.

Using this algorithm, we can obtain the data needed to render caustics, cast by a number of light sources with given spectral characteristics. For each such light source, a spectrum can be divided into a number of zones, and each one of them is assigned a ray with certain parameters. Passing this set of rays to the ray tracer, we receive back a set of ray-cones, corresponding to beams of light which exited the object after a number of internal reflections and splittings. An entering ray-cone is shaped in a form of corresponding model side. Due to dispersion and multiple internal reflections it is split into a number of sub-cones. When one of the sub-cones exits the model, the outline of a cone-model intersection forms a polygon which we call a virtual facet. Thus, each exiting ray-cone corresponds to a specific wavelength and is shaped by a virtual facet, which, quite obviously, also determines the shape of a single-wavelength caustic cast on a plane. Obtained data can be split into a number of ray-cone batches, where in each batch all the ray-cones have originated from the same entering cone and have different wavelengths and intensities. Such a batch of ray-cones forms a "rainbow" of caustic rays. As you can easily figure out, for models with planar sides all cones from a single batch intersect a plane in a straight line, if any, due to the nature of dispersion.

4. CAUSTICS RENDERING

4.1 Projecting Caustics on Plane

After we got a ray exit positions and directions (a ray-cone), we can project this beam onto desired surface. Due to rendering pipeline specifics, projection is done in the view space.

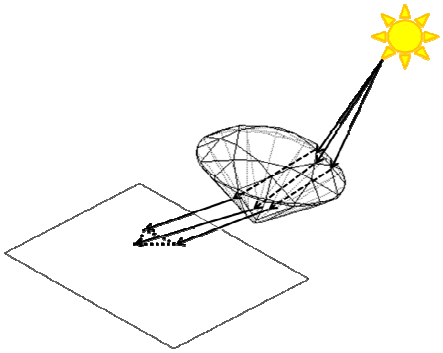

Figure 2: Tracing a ray and projecting a ray cone on a plane.

After we have applied the knowledge about the beam brightness falloff due to refractions and reflections, we should take in consideration the square actually lit by the particular caustic. This is described in the next subsection.

4.2 Using Size Estimation for Correct Lighting

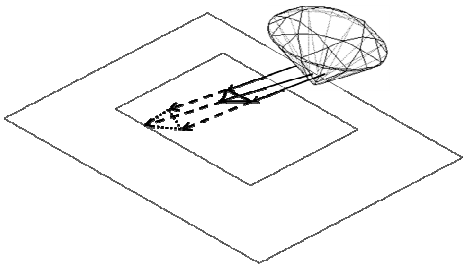

Figure 3: Projecting the same ray cone on two parallel planes.

As could be seen from Figure 3, square of the same ray cone's projection on parallel planes can vary greatly. The cosine between rays and plane normal is constant so we cannot use it to estimate the square. Instead we estimate the square of the ray cone section and its projection using Geron's equation.

Another problem (which hasn't yet been solved) is that the light intensity is non-uniformly distributed across the streak. We are going to solve this in future by means of per-pixel lighting with GPU.

5. SMOOTH CAUSTICS

When one traces rays, corresponding to finite (low) number of wavelengths, problem of gaps between caustics, corresponding to different wavelengths but same vertex can occur. Increasing the number of wavelengths, traced per vertex, reduces frame rate dramatically. Besides, a problem of visible gaps between ray-plane hits occurs even for the case of 20 rays. So, an adaptive solution is needed, which process correct caustics in both cases when different wavelengths hit the surface close to each other and when they are at the considerable distance from each other.

5.1 Interpolation and Integration

First, let's consider a case of three rays (R, G, B). It is easily generalized on the case of more rays.

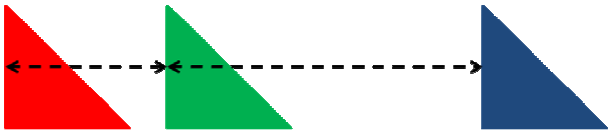

Figure 4: Interpolation of spectrum caustics. Arrow ends are pointing to the point where color is considered to be R, G or B.

As shown in Figure 4, color is interpolated between pairs of neighboring wavelengths. Unfortunately, these values couldn't be used straightforward. One of the possible reasons is shown in Figure 5.

Figure 5: Interference of light between different wavelengths.

Consider the image in Figure 5. Due to the interference of different triangles, resulting energy in each point may vary

according to the distribution of caustics, corresponding to different wavelengths on the surface. We cannot achieve this effect using interpolation only (e. g. we can never obtain white light, even if caustics are directly mapped onto each other).

We propose to integrate these interpolation results in each point to solve this problem. In Figure 6 you can see the set of polygons which contribute to the yellow point (utmost left and right triangles are shown in dashed).

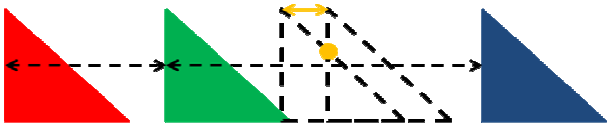

Figure 6: Integration of interpolation results.

In the current realization, integration is performed via additive blending. We render all the interpolating triangles with a given step (the step is taken in the coordinates of the view space) and divide their brightness by the number of actually drawn triangles. The results are shown in Figure 7.

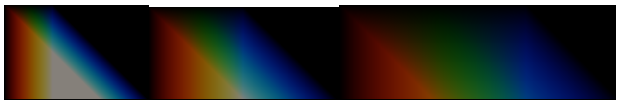

Figure 7: Results of caustics interpolation. From left to right, the distance between caustics is increasing.

5.2 3-Ray Energy Conservation Problem and Solution

While interpolating between red-green and green-blue wavelengths, another artifact occurs. When you integrate these results together, it turns out that the green component is two times brighter than the red or the blue one (because it is actually added twice).

We solve this problem by adding virtual black caustics 'before' red one and 'after' blue one. We extrapolate position of red and blue components using Equation 1 and 2:

$$P_{blue-black} = 2P_{blue} - P_{green} \quad (1)$$

$$P_{red-black} = 2P_{red} - P_{green} \quad (2)$$

where P stands for position of corresponding caustics.

As a matter of fact, the same problem arises when more rays are traced, but it is practically undetectable in that case, because boundary wavelengths make a very low-weight (approximately $1/20^{th}$ of the energy) contribution to the brightness (in case of 20 rays) or even less (due to their proximity to ultraviolet/infrared color).

6. CONCLUSION AND FUTURE WORK

6.1 Results

We have developed a method for rendering smooth spectrum caustics on plane surfaces. Although it works for arbitrary number of rays and produces accurate results interactively, we have vast plans for optimization and enhancement of this technique.

Integration using additive blending is extremely time-consuming. It is possible to evaluate this integral analytically if a linear interpolation is supposed. In this case, color would be a square function of coordinate. This makes it possible to shade each caustic with a pixel shader, which needs to evaluate only a square function in each fragment.

Caustics are usually produced by a volume light source, rather than a point one. That is why they shouldn't be as rough-edged as they are now. The blurriness of the caustics depends on the light source size, the distance between caustics emitter and the plane and the distance that light had actually travelled within the solid object during multiple refractions and reflections. In our ray tracing model, these lengths could be accurately evaluated, which makes it possible to apply a physically-based blurring of caustics.

7. REFERENCES

[1] Chris Wyman. *Interactive Refractions and Caustics Using Image-Space Technique. In ShaderX^5: Advanced Rendering Techniques, edited by Wolfgang Engel, pp. 359-371, Charles River Media, 2006.*

[2] Masahiko Nitanda. *Real-time caustics by GPU. In ShadeX^4: Advanced Rendering Techniques, edited by Wolfgang Engel, pp. 201-210, Charles River Media, 2006.*

[3] László Szirmay-Kalos, Barnabás Aszódi, István Lazányi. *Ray Tracing Effects without Tracing Rays. In ShaderX^4: Advanced Rendering Techniques, edited by Wolfgang Engel, pp. 201-210, Charles River Media, 2006.*

[4] Juan Guardado and Daniel Sánchez-Crespo. *Rendering Water Caustics. In GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Rendering, edited by Randima Fernando, pp. 31-44, Addison-Wesley Professional, 2004.*

[5] Kei Iwasaki, Fujiichi Yoshimoto, Yoshinori Dobashi, Tomoyuki Nishita. *A Rapid Rendering Method for Caustics Arising from Refraction by Transparent Objects. In Proceedings of the 2004 International Conference on Cyberworlds (CW'04), 2004.*

[6] Jong Seo Kim, Kang Soo You and Hoon Sung Kwak. *Caustics Effects with Photo-Realistic Rendering on Movie ('Cars'). In Proceedings of Fifth International Conference on Software Engineering Research, Management and Applications, pp. 274-278, 2007.*

[7] Musawir A. Shah, Jaakko Konttinen, and Sumanta Pattanaik. *Caustics Mapping: An Image-Space Technique for Real-Time Caustics. In Proceedings of IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 13, NO. 2, pp. 272-280, MARCH/APRIL 2007.*

[8] Xuan Yu, Feng Li, Jingyi Yu. *Image-space Caustics and Curvatures. In Proceedings of 15th Pacific Conference on Computer Graphics and Applications, pp. 181-188, 2007.*

[9] Wei Hu and Kaihuai Qin. *Interactive Approximate Rendering of Reflections, Refractions, and Caustics. In Proceedings of IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 13, NO. 1, pp. 46-57, JANUARY/FEBRUARY 2007.*

[10] Chris Wyman, Charles Hansen, Peter Shirley. *Interactive Caustics Using Local Precomputed Irradiance. In Proceedings of Proceedings of the 12th Pacific Conference on Computer Graphics and Applications (PG'04), 2004.*

[11] POV-Ray ray-tracer. http://www.povray.org/

About the Authors

Peter Sikachev is a 4th-year student at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, interactive visualization and GPU programming. His contact e-mail is psikachev@graphics.cs.msu.ru.

Ilya Tisevich is a PhD student at Moscow State University, Department of Computational Mathematics and Cybernetics. His contact email is itisevich[a_t]graphics.cs.msu.ru.

Alexey Ignatenko is a researcher at Computational Mathematics and Cybernetics department of Moscow State University. His research interests include photorealistic 3D rendering, 3D modelling and reconstruction, image-based rendering and adjacent fields. His contact e-mail is ignatenko@graphics.cs.msu.ru.

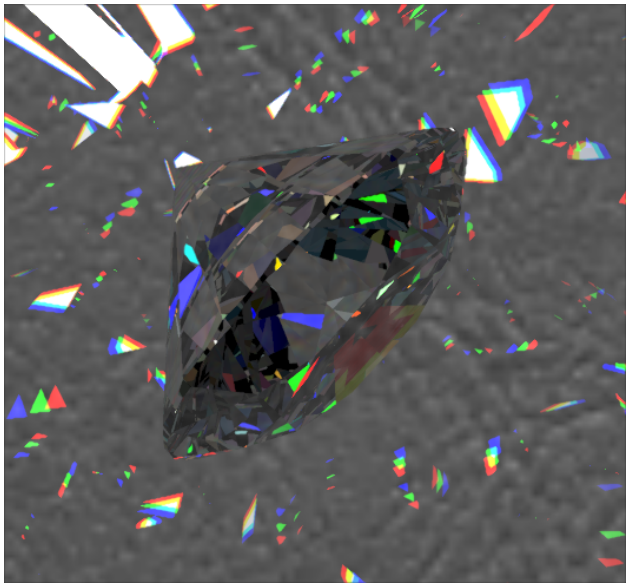

Figure 8: 3-ray discontinuous caustics. Note the discontinuities between R, G and B wavelengths.

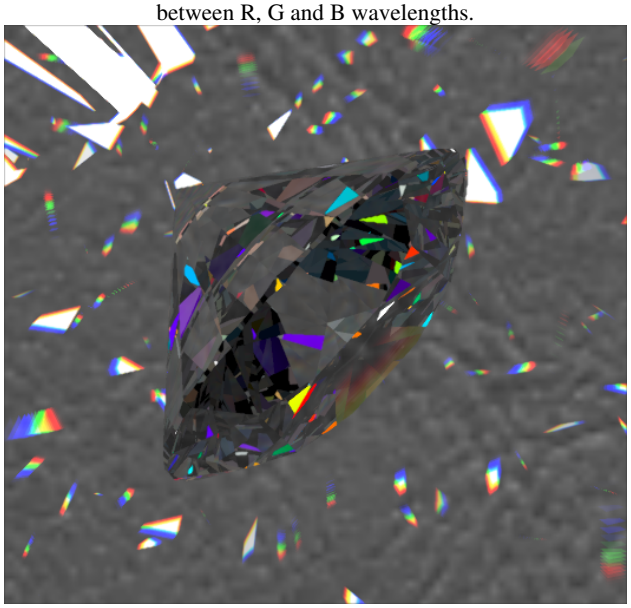

Figure 9: 20-ray discontinuous caustics. Note the discontinuities on the red-green caustic in the top left corner.

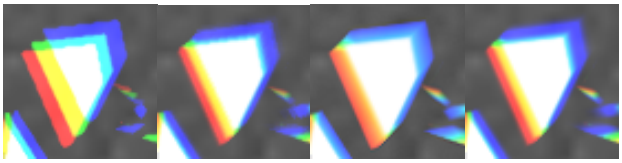

Figure 12: Close-up of single caustic. Quality of rendering is in the same order as Figures 8-11 go. Note the sharp alias border on the most left picture and difference on borders of caustic between 3-rays and 20-rays versions. Difference between 20-rays smooth and sharp version is practically invisible due to close distribution of wavelengths along the surface.

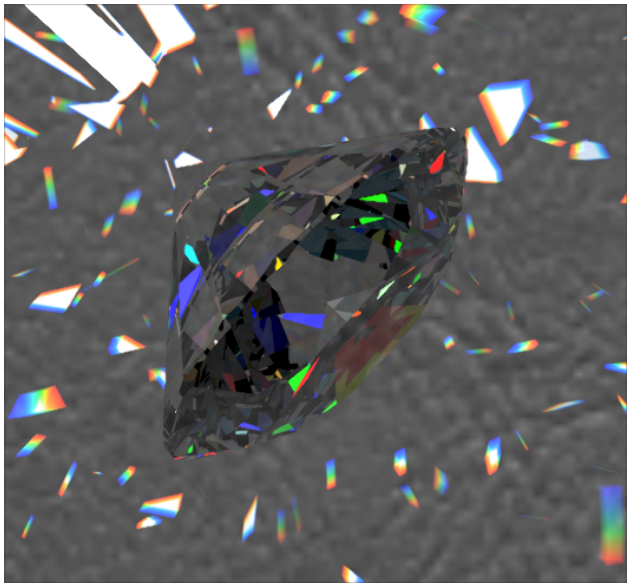

Figure 10: 3-ray smooth caustics. Note absence of discontinuities comparing to Figure 8.

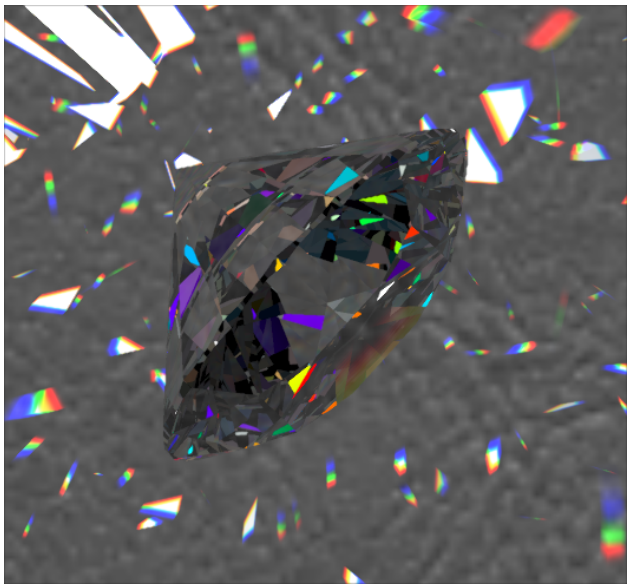

Figure 11: 20-ray smooth caustics. Note the difference with Figure 10 – e. g. the blue color is darker due to more precise calculations.

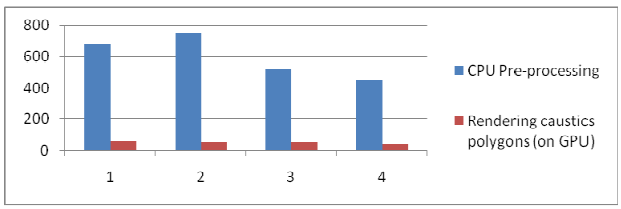

Figure 13: Time of caustics rendering (given in milliseconds, 528×491, camera orthogonal to paper). As could be observed, rendering itself is very fast, so more work should be done on optimizing the pre-processing.

Test configuration: Dell Inspiron 1520, Intel Core 2 Duo T7500 (2.2 GHz), GeForce 8600 M GT, 2 Gb RAM