

RAYTRACING OF DISPERSION EFFECTS IN TRANSPARENT MATERIALS

Alexander Wilkie, Robert F. Tobler, Werner Purgathofer

Institute of Computer Graphics, Vienna University of Technology

Karlsplatz 13/186/2, A-1040 Wien, Austria

e-mail: {wilkie,rft,wp}@cg.tuwien.ac.at

ABSTRACT

In this paper we present a stochastic add-on extension to standard raytracing which makes it possible to render dispersion effects more efficiently than hitherto possible. Our method incurs less overhead than previous proposals, is physically accurate and is applicable to most of the currently feasible spectral representations.

Keywords: Spectral rendering, dispersion, raytracing

1 Introduction

Dispersion, the effect whereby glass prisms and correctly polished clear gemstones break white light into its rainbow components, has fascinated people for ages, and for a long time baffled the attempts of early physics at describing its cause. Eventually, researching this problem led to the first deeper insights into the nature of light, which in turn paved the way for improvements in optics. Nowadays, both its qualitative and quantitative aspects are well understood, and the specific behaviour of all relevant materials can be found in chemical tables.

This knowledge and data is, amongst other areas, routinely used in optical engineering, where a standard problem is the minimisation of the occurrence of this phenomenon in optical systems in order to minimise colour distortion. Manufacturers of gemstones and crystals, on the other hand, are usually intent on maximising the effect, and hence the colourfulness, of their products. The visually pleasing nature of rainbow colours makes crystals popular decorative objects in many contexts of everyday life (from chandeliers to paperweights), and to the accurate observer even numerous “normal” transparent objects like a glass of water or white wine exhibit noticeable colour fringes in their caustics and refractions.

Since the ultimate goal of photorealistic rendering is to generate images that are indistinguishable from reality, one certainly cannot ignore these effects. However, while the principle of the physical phenomenon is simple, its efficient and accurate reproduction is by no means easy.

A solution to this problem was suggested as early as 1986 by Thomas [12], and Collins demonstrated the

ability of his wavefront tracking system to correctly render such effects [1] [2]. Rougeron and Peroche [10] did a rather theoretical study on adaptive representations for spectral rendering which focused on controlling the faithfulness of colour representation, and was less concerned with the technical issues of spectral rendering. Evans and McCool [3] performed a study on efficient spectral rendering with Monte Carlo rendering systems. Their system is capable of generating results that contain dispersion and other effects associated with spectral rendering.

Most of the later work is based on sophisticated, modern rendering algorithms, and are not directly applicable to “plain” raytracing. Based on our experiences with spectral rendering systems, we suggest a simple approach that, while being an improvement over previous similar extensions, is applicable to conventional raytracing, and makes it possible to render dispersion effects without having access to a genuine Monte Carlo renderer.

2 Dispersion in Dielectric Materials

Dispersion occurs where polychromatic light is split into its spectral components on a refractive material boundary, due to the fact that the index of refraction in transparent materials is dependent on the wavelength of the incident light. To complicate matters, this dependency on wavelength is non-linear and related to material constants that have to be measured in experiments.

As one can infer from their near total absence in practical treatises on optical engineering, physically exact analytic descriptions of the dispersion behaviour of real-world materials are highly complex, and are neither necessary nor feasible for engineering purposes.

What is instead routinely used instead by optical engineers and physicists alike, and what is also sufficient for the needs of computer graphics, are empirical approximation formulae that are valid for the limited range of the visible spectrum.

2.1 Sellmeier Coefficients

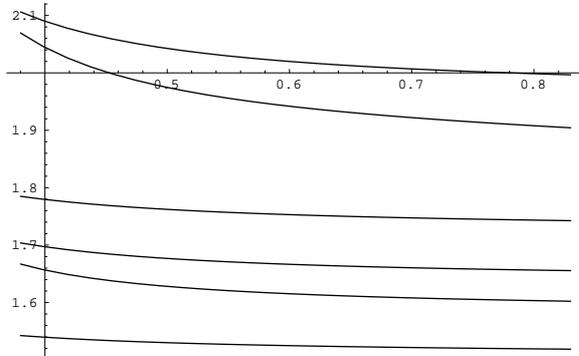


Figure 1: refractive indices for several materials. From top: diamond, lead crystal and several other glass types. Notice the varying amount of dispersion and non-linearity for different materials.

The most widely used method of specifying the dispersion curve for materials in the visual range is to use the so-called *Sellmeier approximation*. Several basically similar forms exist that differ only in the number of empirical constants in structurally similar equations. The number of these constants usually depends on the measurement process by which the data for the approximation is obtained.

A typical example is the glass catalog of the company Schott Glaswerke, which is one of the worldwide leading suppliers of technical glass. In the catalog the technical data of the several hundred types of glass that the company sells is listed, and for specifying dispersion the form

$$n^2(\lambda) - 1 = \frac{B_1\lambda^2}{\lambda^2 - C_1} + \frac{B_2\lambda^2}{\lambda^2 - C_2} + \frac{B_3\lambda^2}{\lambda^2 - C_3} \quad (1)$$

is used, where n is the index of refraction at wavelength λ . The catalog lists coefficient values of B_n and C_n for the different glass types (ranging from normal window glass to highly dispersive lead crystal). In this particular case one can compute the index of refraction for wavelengths from ultraviolet to far infrared with a relative error of less than $1.0E-5$ from just six coefficients per glass type. This makes the catalog a valuable source for accurate dispersion data, especially since it can be downloaded from the company website free of charge and contains specimens of all the main basic glass types (*i.e.* flints, crowns, lead crystal aso.).

There are also numerous other sources of similar freely available material measurements where one can obtain measurements of genuine materials other than glass (*e.g.* diamonds and other gemstones), both on the web and in book form.

3 Spectral Raytracing

The central algorithm of photorealistic rendering is raytracing in its various incarnations. Conventional rendering systems directly use tristimulus values, in most cases device-dependent RGB values, although the device-dependency of such values is often ignored. More sophisticated systems use a device-independent colour space (at least for internal calculations), such as CIE XYZ or CIE L*u*v.

Genuine spectral rendering systems are still rare. They are more difficult to implement, consume significantly more system resources when running, are slower and they require real-world measurements of materials (which are not easy to obtain) as colour input.

While these disadvantages are very obvious, their advantages are for most purposes too obscure to warrant the extra effort for their implementation. This is even more the case since most (if not all) the effects that can only be “genuinely” reproduced through spectral rendering (such as rainbows) are rarely needed for normal CG work and can, in the rare case when they are indeed indispensable, be faked by a good 3D artist in conventional systems at a fraction of the computational cost.

The big advantages of spectral rendering are colorimetric accuracy and the possibility of using physically-based approaches for the reproduction and treatment of various optical and natural phenomena such as *e.g.* metamerism, dispersion, phosphorescence and fluorescence [4] or bioluminescence [6].

3.1 Spectral Representations

One of the key problems of spectral rendering systems is what kind of internal representation one chooses for the spectra that take the place of colour values used in conventional rendering systems. While determining object intersections used to account for the majority of CPU time spent in simple raytracing, the rise of complex surface models (*e.g.* those that account for BRDFs in a proper way) has greatly increased the significance of efficient colour computations in the design of a rendering system.

There are several possible ways of storing spectral data, all of which have to make a space-accuracy-efficiency tradeoff. The three key properties of spectral representations are memory efficiency, accuracy

of reproduction and the cost of colour computations when using a given form of storage.

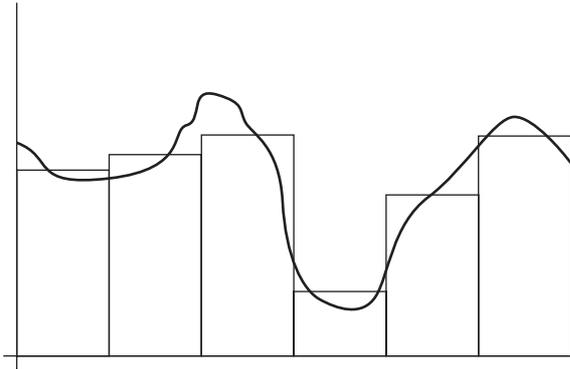


Figure 2: Continuous spectrum sampled by 7 evenly spaced bands.

Sampling

The obvious choice of choosing n appropriately spaced samples (as *e.g.* shown in figure 2) across the visible spectrum has a number of disadvantages, but is still the best choice for a general rendering system. A problem is that one needs a considerable number of samples if reasonable accuracy is desired; for normal rendering, 8 or 16 samples seem to suffice in almost any case, but for high colorimetric accuracy and reference solutions up to 45 or even 100 samples can be necessary. Also, such a sampling approach is naturally prone to aliasing problems in the spectral domain.

On the other hand, provided that the same sampling is used, colour operations on such spectra are very efficient; multiplication of two colour values, which is a (in the general case rather costly) folding operation between the spectra involved, is reduced to a pairwise multiplication of matching coefficients. This, their constant size, and the still reasonable accuracy one can obtain with comparatively few samples (*cf.* the comments of Hall [5] about the benefits of suitable sample spacing) make them the prime choice as spectral representations for a real-world rendering system.

Basis Functions

Peercy [9] proposed using sets of basis functions for representing spectra. This method, while having the nice property of requiring comparatively few coefficients for a potentially highly accurate representation, is problematic because no single small set of basis functions is suitable for all possible spectra. One has the choice of allowing separate bases for each spectrum (this makes colour arithmetic, *i.e.* the integral folds, very costly), or choosing a basis that has few coefficients, but misrepresents certain spectra, or hav-

ing so many basis functions that the advantage of having to store few coefficients is eroded. Also, sets of few, smooth basis functions are not very good at representing spectra with spikes or marked irregularities, which are common in many lightsource spectra.

Hybrid

Based on the observation that reflection spectra are usually smooth (and hence easy to represent through a basis function approach), and that only light source spectra may contain problematical peaks, Sun [11] suggested a hybrid scheme where the low-frequency shape of the spectrum and the spikes of an illumination spectrum are stored separately.

With respect to memory usage, this approach is a very efficient way of representing spectra, but it has the drawback of requiring more complicated folding operations than just pairwise multiplication of coefficients, since the added-on spikes require special treatment.

4 Raytracing Dispersion

The inclusion of dispersion in a rendering system brings about an effect which “standard” deterministic raytracing cannot handle: that a previously sharply defined primary ray suddenly fans out across a solid angle.

Such an effect is readily tractable by spectral versions of Monte Carlo raytracing methods, such as path tracing, or metropolis light transport. However, due to the high computational effort, the slow convergence and the pixel noise associated with these approaches, it would be interesting to have an extension to “normal” raytracing that still enables us to see dispersion at lower computational cost, *i.e.* in the context of an otherwise deterministic raytracer.

This is a feasible proposition because, unlike many of the other problems (such as arbitrary BRDFs) that require the introduction of Monte Carlo into raytracing methods on a general scale, the effect of dispersion is isolated to the ray fan-out at the object boundaries where the split into its spectral components occurs (see figure 3 for an illustration).

4.1 Previous Work

Thomas [12] proposed just such an addition for a normal raytracer. Rays that hit a refractive object boundary are adaptively split into subrays, depending on the solid angle the dispersion causes and the distance the split rays have to travel until they hit the next surface (see figure 3 for an illustration). This approach yields satisfactory results in most cases, but it has drawbacks.

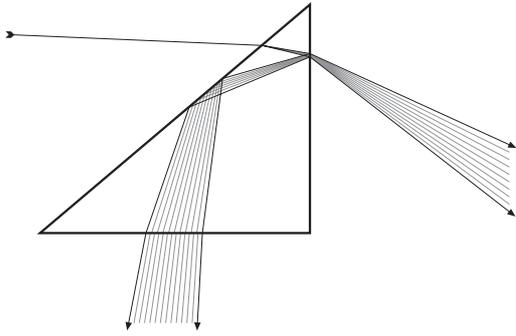


Figure 3: Incident ray being split by a prism with adaptive, deterministic addition of sampling rays: a potential drawback of the method proposed by Thomas [12] is that the subdivision generates unnecessarily large amounts of evenly-spaced (and hence aliasing-prone) rays.

Due to the fact that for reasons of efficiency one has to put a limit to the adaptive splitting of the rays, it starts to fail if a large number of interreflections (*e.g.* inside a crystal) occurs. Also, for large dispersion angles it generates a very large number of sampling rays, since the fan-out of the method at each refraction is theoretically very large. Since ray-scene intersections are the single most costly operation in raytracing, these rays perform dense sampling of the dispersion, but are expensive to trace and are (due to their origin in a regular subdivision) still prone to aliasing artifacts. This makes the method accurate, but slow.

However, with high subdivision levels this method is well suited for the generation of reference images with a purely deterministic raytracer.

5 Our Proposal

Instead of the adaptive approach favoured by Thomas, we propose a simple stochastic sampling scheme which, with minor adaptations, theoretically also works for a system that uses a basis function approach to storing spectra (although the hybrid approach of Sun [11] would pose problems), and that has the advantage of keeping a bound on the number of rays cast from a dispersive surface. Moreover, even the straightforward deterministic version of this technique is capable of producing reasonably good images if the dispersion effects are small, such as the colour fringes in crystal glasses on a table. Even figure 4 was rendered using the deterministic approach, and hardly any artefacts are visible. Only when one zooms in do banding problems start to emerge.

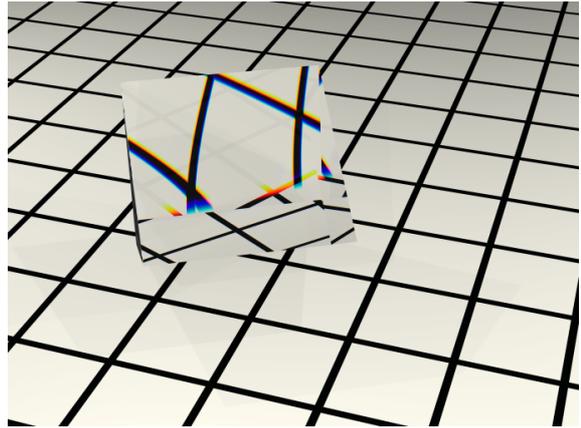


Figure 4: Glass prism floating over a grid-textured plain. The following test images are enlargements of the large lower left texture refraction in the prism.

Deterministic Case

If spectra are represented by n non-overlapping spectral bands, a deterministic raytracer, on encountering a refraction with dispersion, shoots n different rays, each with only the colour contribution of channel n different from zero, in the direction corresponding to the average wavelength of the band. This reduction to n frequencies, and hence n discrete angles of refraction, leads to geometric aliasing, which (as demonstrated in figure 5) can pose a serious problem when dispersion effects are closely viewed (such as in figure 8). However, for images that exhibit only small dispersion effects, like *e.g.* coloured fringes in glasses, this approach is perfectly valid.

The fan-out on subsequent refractions is 1 in this case, since for each band a monochromatic ray, which cannot be split any further, is propagated. The maximum increase in computation time is n -fold for each ray that enters a dispersive medium.

Stochastic Case

The only difference to the deterministic case is that the contribution of each of the n spectral bands is randomly jittered by as much as half the width of the band.

Due to the spreading of each contribution across one neighbouring band, the maximum fan-out in this case is 2, which in conjunction with multiple interreflections could lead to a considerable slowdown compared to the deterministic version, especially when nested interreflections are viewed. However, in practice we found the rendering times to be only up to two times slower than in the deterministic case, indicating that the increase in fan-out affects only the first refraction.

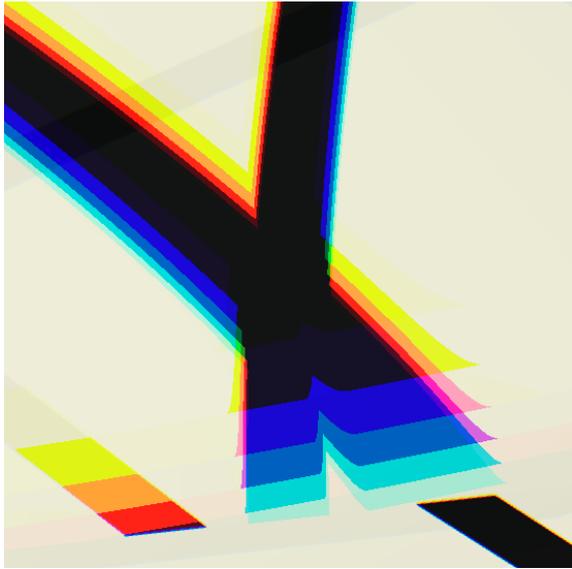


Figure 5: Aliasing artifacts with a spectral representation of 8 regularly spaced samples (enlargement from figure 4). The banding effect almost completely destroys any perception of the colour fringes.

6 Results

We implemented the proposed dispersion sampling scheme in the Advanced Rendering Toolkit (ART), a modular object-oriented graphics framework developed by our Institute which serves as our rendering research platform and resides in the public domain.

ART is currently able to perform colour computations using spectral representations with 8, 16, 45 and 450 samples (the latter exists only for computing of reference solutions), although other choices could easily be added. The images for this paper were rendered using the spectra with 8 samples, because defects in dispersion tracing would show more prominently if the samples are spaced widely apart.

The resulting images were stored in the high-dynamic range Log L*u*v TIFF format proposed by Greg Ward Larson [7] and tonemapped using the interactive calibration method proposed by Matkovic [8]. Also, a local hue-preserving gamut mapping method was performed on out-of-gamut pixels. The last step was necessary since pure spectral colours such as those caused by dispersion are usually outside the gamut of conventional displays.

We deem the artefacts incurred through stochastic sampling, as shown in figure 6, to be less problematic than those that result from the deterministic approach (shown in figure 5), especially since an adaptively sampling raytracer smoothes out most of this dispersion noise through supersampling (see figure 7).

The two test images we present in this paper both



Figure 6: One primary ray per pixel, again with a spectral representation of 8 regularly spaced samples (enlargement from figure 4). This image clearly exhibits the jittering noise introduced by the stochastic sampling. However, at the same rendering speed, the noise artefacts in this image are less disturbing to the perception of the dispersion-induced colour fringes than the colour banding in figure 5.

show transparent objects with the dispersion characteristics of real glass types (normal window glass for the prism, lead crystal for the paperweight in figure 8), but no light absorption within; the latter omission was done on purpose in order to maximise the dispersion-induced colour fringes on refractions for demonstration purposes. Real objects would not be as colourful, as light absorption in the glass cancels out a substantial amount especially of higher-order internal reflections.

Timings

The enlargement shown in figures 5, 6 and 7 is a good benchmark for rendering times because all primary rays hit a dispersive medium. The rendering times are for a single ray per pixel (in order to avoid an unpredictable biasing by the adaptive sampler), with 8 spectral channels in a resolution of 360 by 226 pixels on a PIII with 450MHz.

Non-dispersive rendering of the cut-out takes 70 seconds. The deterministic dispersion tracer takes 540 seconds of CPU time. This is consistent with the predicted slowdown by a factor of n , where n is the number of spectral samples. The stochastic dispersion tracer takes 1133 seconds; this is again consistent with the additional slow-down by a factor of 2 due to the spreading of the channel contributions across two (instead of one) rays.



Figure 7: Up to nine primary rays per pixel, adaptively sampled, 8 spectral samples. (enlargement from figure 4). Note that while some spectral noise remains even here, the overall impression is already satisfyingly smooth.

As we found in other, similar, test runs with different scenes, such timings are typical of the speed loss incurred by dispersion tracing in ART.

7 Conclusions and Future Work

We have presented a simple extension to standard raytracing that provides the possibility of rendering dispersion effects in the context of an otherwise deterministic standard raytracer at calculable and (given the nature of the problem) tolerable expense.

In the future, we intend to investigate the effects of stochastic sub- and supersampling, monochromatic jittered rays (in order to reduce the fan-out) and different jittering probabilities on the convergence properties of the algorithm.

Acknowledgements

The authors would like to thank Laszlo Neumann, Jan Pfikryl, Thomas Theußl and Gerd “Nerd” Hesina for their valuable comments hints and internal reviews. This work has been supported by the “Hochschuljubiläumsfonds der Österreichischen Nationalbank” under project number 7441.

References

[1] S. Collins. Rendering crystal glass. Technical Report TCD-CS-94-20, Department of Computer Science, University of Dublin, Trinity Col-

lege, 31 Dec. 1994. Fri, 18 Apr 1997 12:38:03 GMT.

- [2] S. Collins. *Wavefront Tracking for Global Illumination Solutions*. PhD thesis, Trinity College, Dublin, Ireland, Nov. 1996. Available from <http://isg.cs.tcd.ie/scollins/work.html>.
- [3] G. F. Evans and M. D. McCool. Stratified wavelength clusters for efficient spectral monte carlo rendering. In *Graphics Interface*, pages 42–49, June 1999.
- [4] A. Glassner. A model for fluorescence and phosphorescence. In *Fifth Eurographics Workshop on Rendering*, pages 57–68, Darmstadt, Germany, June 1994. Eurographics.
- [5] R. Hall. Comparing spectral color computation methods. *IEEE Computer Graphics and Applications*, 19(4), 1999.
- [6] B. Hamburg, J. Mogul, B. Reid, A. Eustace, R. Swan, M. Doherty, and J. Bartlett. Characterization of organic illumination systems. Technical Report TN 13 (1989), Digital Equipment Corp. Western Research Laboratory, 1989.
- [7] G. W. Larson. Logluv encoding for full-gamut, high-dynamic range images. *Journal of Graphics Tools*, 3(1):15–31, 1998.
- [8] K. Matkovic and L. Neumann. Interactive calibration of the mapping of global illumination values to display devices. In *Proceedings of the Twelfth Spring Conference on Computer Graphics*, Comenius University, Bratislava, Slovakia, June 1996. Available from <http://cg.tuwien.ac.at/wp/SCCG96-proceedings>.
- [9] M. S. Peercy. Linear color representations for full spectral rendering. In J. T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 191–198, Aug. 1993.
- [10] G. Rougeron and B. Péroche. An adaptive representation of spectral data for reflectance computations. In *Eurographics Rendering Workshop 1997*, pages 127–138, St. Etienne, France, June 1997. Eurographics.
- [11] Y. Sun, F. D. Fracchia, and M. S. Drew. A composite model for representing spectral functions. Technical Report TR 1998-18, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, Nov. 1998.
- [12] S. W. Thomas. Dispersive refraction in ray tracing. *The Visual Computer*, 2(1):3–8, Jan. 1986.

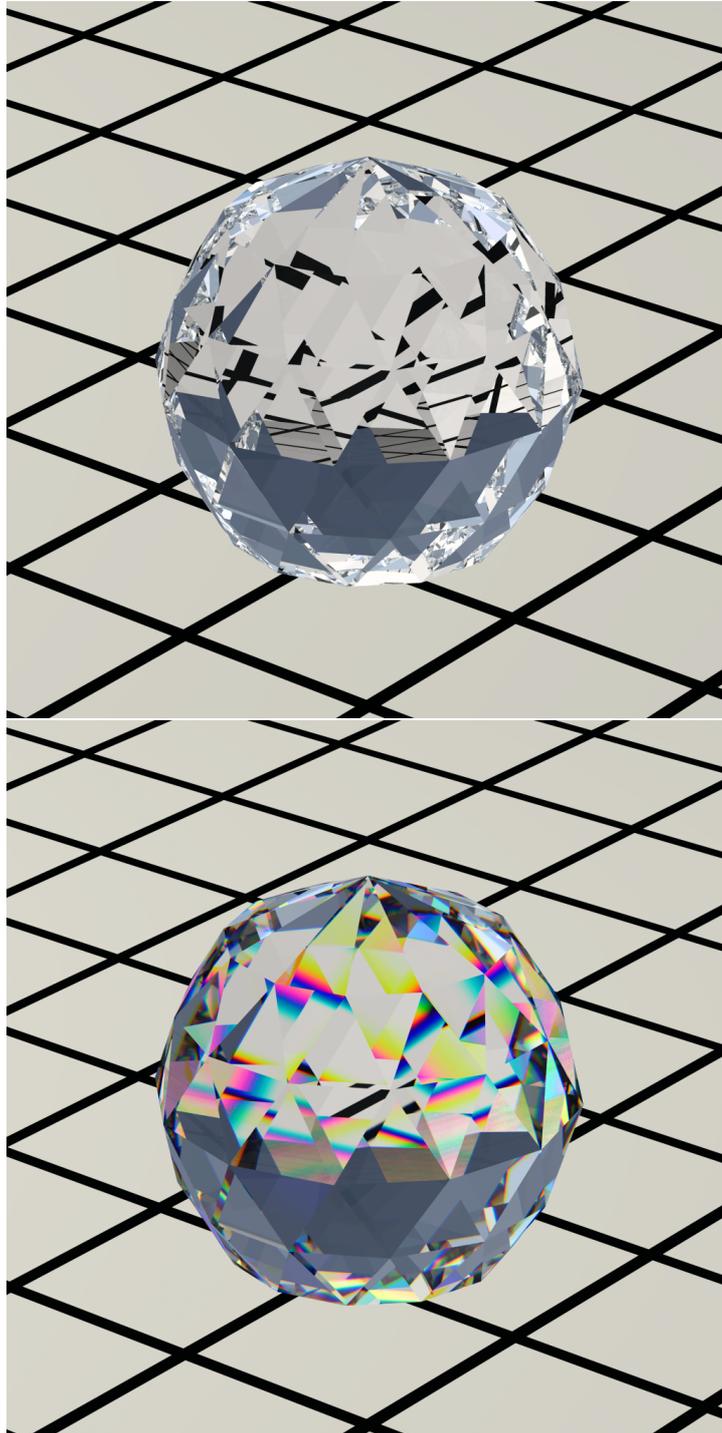


Figure 8: A lead crystal paperweight rendered without and with dispersion taken into account. The lower image exhibits more colourful patterns than such objects normally do because the internal absorption of the glass was set to zero. This was done in order to better demonstrate the ability of our proposed method to reproduce colour gradients caused by dispersion even after nested interreflections.