

# Decoupling Illumination from Isosurface Generation Using 4D Light Transport

David C. Banks and Kevin M. Beason

**Abstract**—One way to provide global illumination for the scientist who performs an interactive sweep through a 3D scalar dataset is to pre-compute global illumination, resample the radiance onto a 3D grid, then use it as a 3D texture. The basic approach of repeatedly extracting isosurfaces, illuminating them, and then building a 3D illumination grid suffers from the non-uniform sampling that arises from coupling the sampling of radiance with the sampling of isosurfaces. We demonstrate how the illumination step can be decoupled from the isosurface extraction step by illuminating the entire 3D scalar function as a 3-manifold in 4-dimensional space. By reformulating light transport in a higher dimension, one can sample a 3D volume without requiring the radiance samples to aggregate along individual isosurfaces in the pre-computed illumination grid.

**Index Terms**—physically-based illumination, isosurface, level set, light transport.

## 1 ILLUMINATION FOR 3D DATA VISUALIZATION

This paper addresses the issue of displaying globally illuminated isosurfaces in real time. Its main contribution is the novel formulation of the light transport equation that can be applied to the graph of a scalar function in  $\mathbb{R}^{n+1}$  to pre-compute illumination for level sets of the function in  $\mathbb{R}^n$ . The idea is motivated by the following observation: level sets of a function can be stacked to produce the graph of the function. For example, a scalar function of two variables has isocurves as its level sets; the isocurves can be stacked to form a surface (the graph of the function) in  $\mathbb{R}^3$ . If the surface is illuminated by a light source, the individual level sets are illuminated as well. Many physically-based (also called “global”) illumination solvers already exist for surfaces in  $\mathbb{R}^3$ , and could be employed to illuminate the continuous stack of isocurves as a byproduct of illuminating the surface that the isocurves fill, or foliate. There is thus no need to construct and illuminate isocurves individually.

By analogy, if a global illumination solver were available for a volumetric 3-manifold in  $\mathbb{R}^4$ , it could be employed to pre-compute the global illumination solution for all isosurfaces collectively, without the need for constructing individual isosurfaces one by one.

Two problems arise with this plan, which we solve in this paper. First, the result of illuminating the surface in  $\mathbb{R}^3$  is actually not quite the same as illuminating each of its level sets in  $\mathbb{R}^2$ , so naively illuminating the surface in  $\mathbb{R}^3$  as described above will not produce the correct results. Second, in order to implement the idea of illuminating the graph for functions of three variables rather than two, it is necessary to develop a plausible model for light transport in  $\mathbb{R}^4$ .

Previous research has addressed aspects of these concerns. The problem of illuminating a 3-manifold in  $\mathbb{R}^4$  has been discussed by Banks [1], Hollasch [8], and Hanson *et al.* [6], although only in the context of diffuse and specular reflection for local illumination, not global illumination.

The basic idea of pre-computing illumination of isosurfaces and caching the result in a texture map has been explored by several people. Stewart demonstrated the idea of texture-mapping pre-computed illumination onto isosurfaces [16]. Beason *et al.* [4] applied photon mapping [11] during the batch processing phase of the pre-computation, then texture-mapped the resulting illumination grid us-

ing the isosurface engine in a commercial visualization system [15]. Parker *et al.* [12] used ray tracing for computing global illumination on isosurfaces, and Wyman *et al.* [18] incorporated spherical harmonics for nondiffuse surfaces, allowing view-dependent display.

The notion of restricting the propagation of electromagnetic radiation to a subspace of lower dimension is currently being explored within theoretical physics. According to one theory, the effect of gravitation extends in many dimensions, but electromagnetism is confined to the 4-dimensional subspace (called a membrane or “brane”) of space-time that we are familiar with [13]. In a similar vein, we show how light transport in  $\mathbb{R}^{n+1}$  can be restricted to an  $n$ -dimensional subspace, and demonstrate the solution for  $n = 2$  and  $n = 3$ . This approach permits an efficient sampling strategy for pre-computing global illumination of isosurfaces to be developed, which can be used for decreasing the error in the resulting illumination grid or for reducing the number of samples required to meet a desired error tolerance.

### 1.1 Local and Global Illumination

Visualizing a three-dimensional (3D) scalar function is an important component of the workflow for visual data analysis in science, medicine, and engineering. Many software systems for 3D data visualization offer only the default rendering capability provided by the computer’s graphics card, which implements a graphics library such as OpenGL at the hardware level to render polygons using local illumination.

Physically-based illumination entails solving the light transport

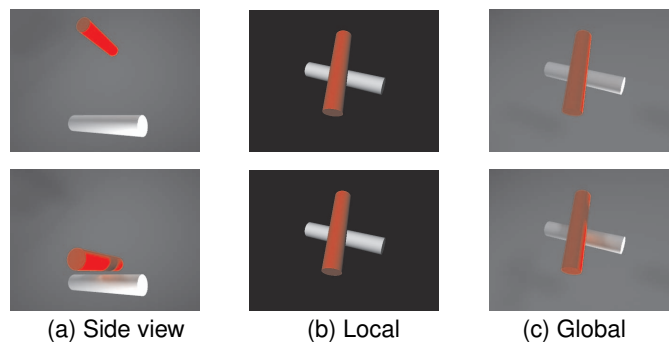


Fig. 1. Global illumination reveals imminent object contact between two cylinders. In the top row the two cylinders are far apart; in the bottom row they are close together. (a) A side view reveals the difference between the two configurations. (b) and (c) When viewed from above, the two poses are difficult to discriminate. (b) Under local illumination, the two configurations are indistinguishable. (c) Shadows and inter-reflection from global illumination convey that the two cylinders nearly touch in the configuration in the bottom image.

- Banks is with the University of Tennessee, the UT/ORNL Joint Institute for Computational Sciences, the ORNL Scientific Computing Group, and the Harvard NeuroDiscovery Center; email: dbanks@eecs.utk.edu.
- Beason is with Rhythm and Hues Studios; email: beason@rhythm.com.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

equation in a scene, which often yields realistic-looking images. The difference between local illumination and physically-based illumination is illustrated in Figure 1, which shows a pair of cylinders in two spatial configurations. In the first configuration (top row) the two cylinders are far apart, separated by several diameters. In the second (bottom row) they are close together, separated by less than one diameter. When the scene is viewed from judiciously chosen positions (Figure 1a) a gap appears between the silhouettes of the surfaces and the amount of separation between the cylinders is evident. But from other viewing positions, their separation distance must be inferred from other visual cues. In Figure 1c shadows and inter-reflections reveal that the surfaces are proximal to each other. Local illumination does not provide these visual cues, making the two configurations indistinguishable in Figure 1b. From a viewpoint like that in Figures 1a and 1b, the proximity of one cylinder to another can be better perceived when shadows and inter-reflection are included, as from global illumination. The total effect is superior to shadows alone or inter-reflection alone [10], which invites the natural speculation that the human visual system is hard-wired to infer 3D shapes from images of globally illuminated scenes.

Recent trials with human subjects found that perception of relative depth in a 3D scene (containing many tube-like surfaces) was improved, as well as perception of aggregate shape, when physically-based illumination was employed instead of local illumination [17]. If physically-based illumination offers superior scene perception, one would expect it to be used widely for visualizing scientific datasets. But this is not the case, as can be seen in the recent proceedings of the most recent (2008) Scientific Discovery through Advanced Computing (SciDAC) meeting. The US Department of Energy's SciDAC project provides US\$60 million to fund high-performance computer simulations in diverse fields of science using world-class supercomputing platforms. The 2008 meeting of the SciDAC principal investigators included experts from a broad range of sciences, and offers a snapshot of the state-of-the-art in computational sciences and a depiction of the way scientists present the results of their simulations. The proceedings [5] include 100 papers describing advances in astrophysics, chemistry, climatology, combustion, fusion, life sciences, materials sciences, nuclear physics, quantum chromodynamics, and applied mathematics, with approximately 800 figures. Roughly 50% (418) of these figures are 1-dimensional (1D) plots, graphing one variable against another or displaying 1D isocurves. Another 15% (134) display a function of two variables using a colormap. Some 15% (128) of the figures can be classified as diagrams or charts, often with edges and nodes that illustrate a process. Roughly 5% (22) of the figures are photographs (such as of experimental equipment). The remaining 20% (153) of the figures depict 3D curves, surfaces, volumes. None of the figures include captions that describe the illumination model applied. But the absence of shadows and interreflection indicates that zero (0) of the 153 3D visualizations in the proceedings employed physically-based illumination to render the scenes; instead, all of the scenes appear to be rendered using either no illumination model or a local illumination model.

It is somewhat surprising to find that the "market penetration" of physically-based illumination in scientific visualization was nearly zero in 2008, especially when one considers that some of the datasets result from faithful simulation of transport/scattering phenomena (and therefore could have incorporated physically-based illumination within the simulation pipeline). Recent advances in hardware and algorithms have brought physically-based illumination within reach for interactive display of 3D geometry, and there is evidence (described above) that perception is enhanced by physically-based versus local illumination. One reason scientists give when asked about their use of local illumination for visualizing 3D data is that physically-based illumination is not available as an option in their workflow. Further inspection shows that the workflow sometimes involves legacy software systems for which the scientist has no budget (or incentive) to replace. Our experience has been that the typical scientist is not demanding that physically-based illumination be available as a real-time rendering option for the desktop or the laptop; indeed, the typical scientist

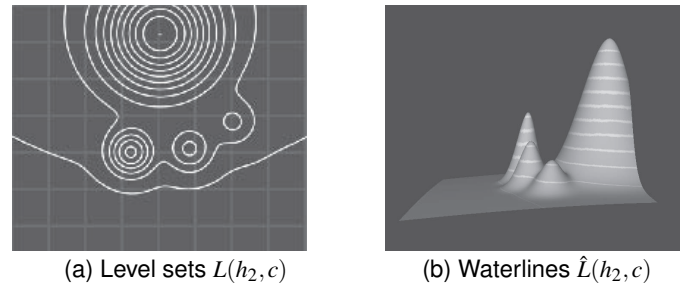


Fig. 2. An example 2D heightfield  $h_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ . (a) Level sets of  $h_2$  for several isovalues  $c$ . (b) Corresponding waterlines, or raised level sets, on the graph  $H_2$ .

is unlikely to use physically-based illumination at all unless it can be employed within the software and hardware toolsets already in place for data analysis. This cultural reality diminishes the usefulness of advances in hardware-supported rendering techniques and places added value on finding ways to introduce contemporary rendering techniques into existing data-analysis workflows.

One way to introduce physically-based illumination into an existing workflow without requiring a change to the rendering hardware or the rendering engine is to cache the pre-computed radiance in a 3D illumination grid, then apply the cached values to isosurfaces at visualization time via ordinary texture mapping. This strategy has been demonstrated using off-the-shelf visualization software without requiring any change to the software system [4] [2]. One drawback to constructing a 3D illumination grid is that constructing and illuminating isosurfaces one at a time requires that the isosurfaces be chosen to uniformly sample the volume they fill. Another drawback with repeated isosurfacing-and-illumination is that the radiance values (computed, for example, using photon mapping) are thereby restricted to isosurfaces and neglect the regions in the volume in between. The following section describes our solution to these drawbacks, permitting illumination to be decoupled from isosurface extraction.

## 2 LIGHT TRANSPORT ON GRAPHS AND LEVEL SETS

A level set  $L(h, c)$  of a scalar function  $h$  is the locus of points  $x$  in the domain of  $h$  such that  $h(x) - c = 0$ . Sweeping through level sets by gradually changing the isovalue  $c$  is a standard way for a user to examine scalar data. For a scalar function of two variables, performing this sweep is like gradually raising the water level with respect to a partially submerged terrain. The height of the terrain is the scalar function (with the terrain serving as the graph of the function); the water meets the terrain at the waterline. When the water rises from a low to a high level, the waterline sweeps out a solid strip of terrain. That is, the terrain equals the union of all its waterlines. A point on the waterline is also a point on the terrain, so one is free to consider a series of waterlines individually or to think of the continuum of waterlines as constituting the terrain.

A level set lies in the domain of the function whereas a waterline lies on the graph of the scalar function. We first consider two-dimensional (2D) scalar functions. Let  $\mathbf{x}_2 = (x_1, x_2)$  be a point in  $\mathbb{R}^2$  (where the subscript of  $\mathbf{x}$  indicates its dimension). The 2D scalar function

$$h_2(\mathbf{x}_2) : \mathbb{R}^2 \rightarrow \mathbb{R}$$

has as its graph the mapping

$$H_2(\mathbf{x}_2) = (x_1, x_2, h_2(\mathbf{x}_2))$$

which forms a surface in  $\mathbb{R}^3$ . The level set  $L(h_2, c)$  lies in the domain  $\mathbb{R}^2$  of the function, and is defined as the locus of points  $\mathbf{x}_2$  satisfying

$$L(h_2, c) = \{\mathbf{x}_2 : h_2(\mathbf{x}_2) - c = 0\}$$

for a given (scalar) value of  $c$ .

The level set lies in the  $(x_1, x_2)$ -plane, not up at the waterline on the terrain. To fit the waterline, the level set must be elevated up from the domain a distance  $c$ , producing a *raised* level set  $\hat{L}(h_2, c)$  along the intersection of the graph  $H_2$  and the plane at height  $c$ . A 3D point  $\mathbf{x}_3$  on a raised level set belongs both to the graph in  $\mathbb{R}^3$  and to the waterline in  $\mathbb{R}^2$ , and can be projected down to the level set in  $\mathbb{R}^2$ .

## 2.1 Flattened Light Transport: $\mathbb{R}^3$ to $\mathbb{R}^2$

Our eventual goal is to illuminate isosurfaces in  $\mathbb{R}^3$  of a function  $h_3 : \mathbb{R}^3 \rightarrow \mathbb{R}$  whose graph is a 3-manifold in  $\mathbb{R}^4$ . But it is useful to first consider the situation one dimension lower where it can be more easily visualized. The lower-dimensional analog would be illuminating an isocurve in  $\mathbb{R}^2$ . Heckbert’s paper “Radiosity in Flatland” describes how global illumination of curves can be performed in the plane [7]. We have the choice of illuminating these level sets  $L(h_2, c)$  down in the plane where they are defined (as in Figure 2a), or illuminating them as raised level sets  $\hat{L}(h_2, c)$  up in the planes  $h_2 = c$  along the waterlines (as in Figure 2b).

For the function  $h_2$ , the graph is a surface in  $\mathbb{R}^3$ . Illuminating a surface in  $\mathbb{R}^3$  is certainly more familiar than illuminating curves in the 2D domain. If one wishes to pre-compute global illumination for many curves in the plane, it is natural to prefer raising them up to the waterline so that they become the graph. In a sense, an ordinary renderer illuminates infinitely many (raised) level sets as a side effect of illuminating a surface. But a subtle problem arises when one imagines computing illumination in this way: light can strike the surface at one height and reflect at an angle up to the surface at another height. This unwelcome behavior is like illuminating one isosurface only to have stray reflections leak over and shine on other isosurfaces.

The remedy we propose is to confine, or flatten, the emittance of the luminaire and the reflectance of the surface so that an “ordinary” renderer will produce an image of the surface *as if* each waterline had been individually illuminated, alone within its plane, just like a level set in flatland.

Scattering, or surface reflectance, can be flattened to coerce photons emitted within a plane to reflect within the plane. The bidirectional reflectance distribution function (BRDF)

$$f : \mathbb{S}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R}$$

at a point on an ordinary surface in  $\mathbb{R}^3$  depends on an incident direction and an exitant direction of the light, where each direction is chosen from a sphere  $\mathbb{S}^2$ . The flattened BRDF

$$f^{\flat} : \mathbb{S}^1 \times \mathbb{S}^1 \rightarrow \mathbb{R}$$

depends on incident and exitant directions chosen from a circle  $\mathbb{S}^1$  within the given plane. The superscript  $\flat$  is the musical “flat” sign.

Flattened radiance  $L^{\flat}$  is the quotient of a 1D length  $\Lambda$  and a 1D direction  $\vec{\theta}$ , *i.e.*, the radiant flux per unit angle per unit projected length:

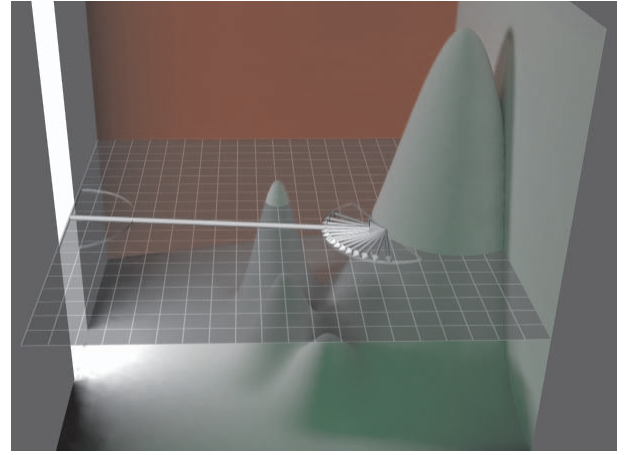
$$L^{\flat}(\vec{\theta}) = \frac{d^2\Phi}{d\Lambda \cos\theta d\vec{\theta}},$$

where  $d\Lambda \cos\theta$  is the projected differential length at position  $\mathbf{x}$  receiving light from differential angle  $d\vec{\theta}$  in direction  $\vec{\theta}$ . The flattened normal  $\mathbf{N}^{\flat}$  is computed by projecting the graph’s normal  $\mathbf{N}$  down to the plane and scaling to unit length. The factor  $\cos\theta$  can be computed as the dot product between  $\vec{\theta}'$  and the flattened normal  $\mathbf{N}^{\flat}$ .

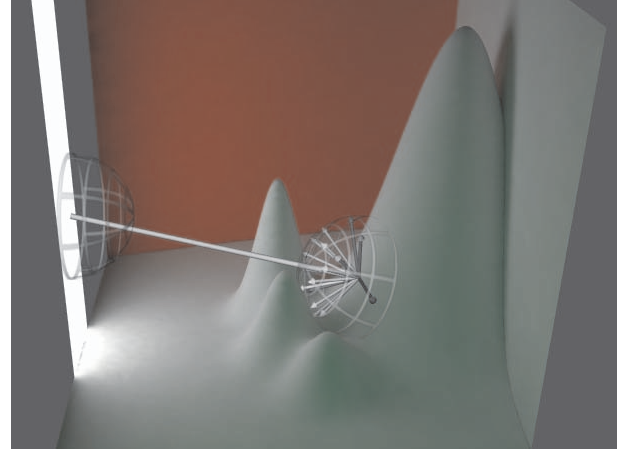
The ordinary rendering equation can be modified to produce flattened light transport by using these flattened terms and by integrating over the circle rather than the sphere:

$$\text{Ordinary } L(\vec{\omega}) = L_e + \int_{\Omega} f(\vec{\omega}, \vec{\omega}') L_i(\vec{\omega}') (\mathbf{N} \cdot \vec{\omega}') d\vec{\omega}' \quad (1)$$

$$\text{Flattened } L^{\flat}(\vec{\theta}) = L_e^{\flat} + \int_{\Theta} f^{\flat}(\vec{\theta}, \vec{\theta}') L_i^{\flat}(\vec{\theta}') (\mathbf{N}^{\flat} \cdot \vec{\theta}') d\vec{\theta}' \quad (2)$$



(a) Flatland illumination in the plane  $h_2 = c$



(b) Ordinary illumination of graph  $H_2$

Fig. 3. Illumination in  $\mathbb{R}^2$  versus  $\mathbb{R}^3$ . (a) A point  $\mathbf{x}_3$  on a raised isocurve  $\hat{L}(h_2, c)$  lies in the plane at height  $c$ , and can be illuminated in 2D flatland. The luminaire emits in the plane, and the point  $\mathbf{x}_3$  scatters within the plane. (b) The same point  $\mathbf{x}_3$  lies on the graph  $H_2$ , which is a surface in  $\mathbb{R}^3$ , and so can be illuminated with conventional 3D lighting. The luminaire emits with a hemispherical distribution in  $\mathbb{R}^3$ , and the point  $\mathbf{x}_3$  scatters light in  $\mathbb{R}^3$ .

where  $L^{\flat}(\vec{\theta})$  is the flattened radiance leaving a point  $\mathbf{x}_3$  in direction  $\vec{\theta}$ ,  $L_e^{\flat}$  is the flattened emittance,  $\Theta$  is the set of directions in the circle around  $\mathbf{x}_3$  in the plane,  $f^{\flat}(\vec{\theta}, \vec{\theta}')$  is the flattened BRDF,  $L_i^{\flat}(\vec{\theta}')$  is the flattened radiance incident to  $\mathbf{x}_3$  from incoming direction  $\vec{\theta}'$ , and  $\mathbf{N}^{\flat}$  is the segment’s normal in the plane. This novel formulation of flattened light transport allows us to either interpret a waterline as an isolated curve in a 2D plane that intersects the graph, and as a portion of the height field in  $\mathbb{R}^3$ . More importantly, the flattened light transport equation can be applied one dimension higher to an isosurface considered as a raised level set in a 3D hyperplane that intersects the graph, and as a portion of the (volumetric) height field in  $\mathbb{R}^4$ .

Flattened light transport may seem purely theoretical, but it has a practical application. Previous work on pre-computing global illumination of isosurfaces either used the underlying data grid for sampling the radiance, or explicitly constructed level sets and illuminated them. The practical value of flattened light transport is that it forces the graph of the scalar function  $h_n : \mathbb{R}^n \rightarrow \mathbb{R}$  to be illuminated within  $n$ -dimensional hyperplanes, just as if the level sets were constructed individually down in the domain. Consequently, a rendering engine for surfaces in  $\mathbb{R}^3$  can be modified so that it illuminates them flatland-style, with no light leaking from one height to another. Pre-computing global illumination of the level sets can therefore be accomplished by

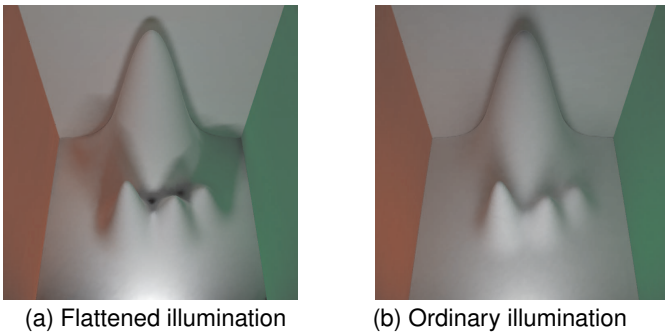


Fig. 4. Comparison of flattened light transport within 2D planes versus ordinary light transport in  $\mathbb{R}^3$ . (a) Flattened transport produces shadows with hard edges, and produces stronger color bleeding from the walls. (b) Ordinary illumination allows light from the luminaire to pass over the small bumps in front so they cast softer shadows on the large bump behind.

assigning flattened emittance to the luminaire and flattened reflectance to the graph, as opposed to implementing a completely different illumination algorithm.

Flattened light transport is illustrated in Figure 3a, which shows the graph  $H_2$  placed in a reflective box together with a rectangular luminaire. The plane  $h_2 = c$  intersects the graph, the box, and the luminaire. Within this plane, the raised level set forms a pair of disconnected waterlines: the small circle cut from a bump near the luminaire, and a larger circle cut from the bump near the wall farthest from the luminaire. The waterlines are surrounded by a square bounding them within the box, with one red edge (on the red wall) and one green edge (on the green wall, which is not shown in the figure so that the interior of the box may be seen; the indirect illumination off the green wall is visible, however, on the graph of the function). The luminaire and the walls of the “Cornell box” are distinct from the graph of the function itself. The walls provide shape cues by capturing cast shadows and by allowing indirect illumination of the scene. The walls can be thought of as extrusions into the third (vertical) dimension of the boundary of the domain.

The waterlines within the intersecting plane are illuminated by a (horizontal) segment of the luminaire lying within the plane. The figure illustrates a photon leaving the luminaire and striking a point  $\mathbf{x}_3$  on the waterline on the far bump of the graph. The semicircle around a point on the emitting segment represents the possible directions in which the photon can be emitted, which are confined within the plane. When a photon strikes a point  $\mathbf{x}_3$  on the waterline, it scatters strictly within the same plane. This is illustrated by a set of arrows leaving the point on the waterline. The flattened illumination can be understood in two ways simultaneously: (1) photons from a 2D luminaire scatter off a waterline within a 2D plane according to the “usual” model for 2D scattering; (2) the 2D luminaire and the waterline lie within a plane in  $\mathbb{R}^3$ , and so the usual model for 3D scattering must be modified to constrain the photons to emit and reflect within the plane.

Figure 3b shows ordinary (not flattened) illumination of the point  $\mathbf{x}_3$ . With ordinary light transport, a neighborhood of  $\mathbf{x}_3$  belongs to a surface (namely the graph) in  $\mathbb{R}^3$  and not just a curve (namely the raised level set) in a plane. The luminaire is free to emit a photon in any direction on a hemisphere, not just directions on a semicircle in the plane. When the photon scatters in  $\mathbb{R}^3$  from the point  $\mathbf{x}_3$ , it is free to follow any exitant direction on a hemisphere aligned with the surface normal. Although the surface  $H_2$  is equal to the union of all of the raised level sets, conventional light transport applied to the surface  $H_2$  differs from flattened light transport applied within each plane.

Figure 4a and Figure 4b show this difference. The image in Figure 4a was rendered with flattened light transport while Figure 4b was rendered with ordinary light transport. With flattened light transport, the shadows are sharper and the color bleeding stronger than with ordinary light transport. Shadows and indirect illumination provide

visual cues for perception of shape in a scene; incorrectly allowing photons to “leak” from sections of the luminaire situated at heights above and below a given waterline, as in Figure 4b, causes lighting artifacts (such as one shape failing to cast a perceptible shadow on another one). These phenomena can be understood based on the diagram in Figure 3a, which shows that restricting light from the luminaire to flow strictly within a plane causes the smaller bumps near the light to block photons from reaching the larger bump behind. Indirect illumination from the red and green walls fills the shadows with pronounced red and green colors. By comparison, light from the luminaire in Figure 3b is able to “leak” out-of-plane over the small bumps and onto the larger bump behind, resulting in a softer shadow being cast, and softening the color reflecting off the side walls. The differences in the flattened illumination and ordinary illumination depend on the geometry of the luminaire and the graph. They are particularly noticeable in the case where the light leaks from the top part of the luminaire over the small bumps (maxima of the scalar function) that lie closer to the luminaire than the large bump on which they cast shadows. The corresponding situation for scalar functions of  $\mathbb{R}^3$  would be that the “usual” process of four-dimensional (4D) scattering would permit photons to leak from one 3D hyperplane to another, thereby introducing different results than for illumination of an isosurface in  $\mathbb{R}^3$ .

## 2.2 Sampling Flattened Radiance on Graphs in $\mathbb{R}^3$

With flattened light transport, the construction and illumination of individual level sets is unnecessary and radiance samples can be decoupled from lattice points of the data. For example, radiance samples can be distributed uniformly, as illustrated in Figure 5c and Figure 5d. In the top row of the figure, 20,000 radiance samples are distributed strictly along level sets. As a result, the interior of the square domain is sampled very non-uniformly. When the scattered samples (Figure 5a) are interpolated onto a uniform 2D illumination grid in Figure 5b, sampling artifacts are evident in the form of bands that follow the pattern of individual level sets. The level sets are oversampled while the regions between them are undersampled. In the bottom row of Figure 5, the 20,000 radiance samples are uniformly distributed in the domain of the function, and sampling artifacts can no longer be seen.

Flattened light transport allows radiance to be calculated at the sample point on the graph itself (in  $\mathbb{R}^3$ ) rather than in 2D flatland. The sampling of the radiance is thereby decoupled from the sampling of the level sets. Not only is the process of generating level sets bypassed, but a given number of samples can be better distributed in order to reduce error in the illumination grid. In the case of the 20,000 samples in the figure, dense sampling of 10 level sets yields a root-mean-square (RMS) error of 7.4% after interpolation onto the 2D grid (using Shepard’s method [14]). Uniformly distributing the 20,000 radiance samples yields an RMS error of 2.2%. Sampling the graph, but illuminating with flattened light transport, thus produces a smaller average error in this example. Generating an optimal pattern of sampling is an area for future research.

## 2.3 Sampling Flattened Radiance on Graphs in $\mathbb{R}^4$

We next consider the higher-dimensional case of scalar functions of three variables whose level sets are isosurfaces in  $\mathbb{R}^3$  and whose graphs are 3-manifolds in  $\mathbb{R}^4$ . In the previous section we described how to flatten 3D illumination by starting with light transport in  $\mathbb{R}^3$  and restricting the emittance, reflectance, and surface normal to a lower dimension. By analogy, to flatten 4D illumination we could begin with “ordinary” light transport in  $\mathbb{R}^4$ . Although it is not obvious how a 4D version of light transport should be written, our goal is merely to illuminate raised level sets lying in 3D subspaces of  $\mathbb{R}^4$ ; thus we simply stipulate that flattened 4D illumination be identical to ordinary 3D illumination when restricted to an isosurface in  $\mathbb{R}^4$ .

For a 3D scalar function

$$h_3(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$$

the level set  $L(h_3, c)$  is the locus of points  $\mathbf{x}_3$  defined by

$$L(h_3, c) = \{\mathbf{x}_3 : h_3(\mathbf{x}_3) - c = 0\}$$

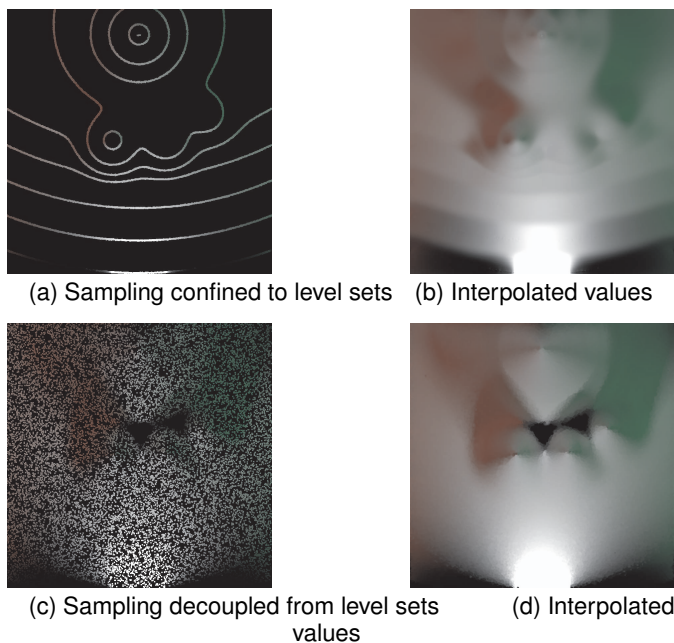


Fig. 5. Sampling and interpolation of radiance samples of the function  $h_2$ . (a) 20,000 radiance samples restricted to level sets. (b) Interpolation onto a high-resolution 2D illumination grid exhibits sampling artifacts, visible as bands that follow the level sets. (c) 20,000 uniformly distributed radiance samples, decoupled from level sets or underlying grid. (d) Interpolation shows no banding.

for a given value of the scalar  $c$ . Physically-based illumination can be (pre-)computed on individual isosurfaces, with the scattered samples of the radiance then interpolated onto a 3D grid for later use as a texture map. Figure 6a illustrates this process for a scalar function  $h_3$  having a three small “bumps” and one large “bump” on its graph in  $\mathbb{R}^4$ . Each isosurface is placed in a 4D Cornell box when illumination is performed. The box contains a red polygonal wall  $p_r$  opposite a green polygonal wall  $p_g$ ; the effect of these walls is visible as color bleeding on the isosurfaces. A polygonal luminaire  $p_l$  shines into the box. The scene is analogous to that in Figure 5a, only one dimension higher. A total of  $10^6$  radiance samples were calculated over several level sets, then interpolated to produce the 3D texture in the middle image of Figure 6a. The rightmost image is an isosurface whose vertices index into that texture map.

In the 4D example, the domain of  $h_3$  is a regular 3D lattice, and the graph of  $h_3$  is composed of cube-shaped hexahedral cells. Let the interval  $I = [h_{min}, h_{max}]$  denote the range of the function  $h_3$ . The polygonal luminaire  $p_l$  is extruded “upward” in the  $h_3$  direction to form a volumetric luminaire  $p_l \times I$  in  $\mathbb{R}^4$ . Likewise, each wall of the box is extruded to form a solid in  $\mathbb{R}^4$ ; for example, the red wall becomes  $p_r \times I$ . The extruded luminaire is allowed to shine onto the volumetric mesh contained within the volumetric box in  $\mathbb{R}^4$ , subject to flattened light transport. Although the 4D scene may be difficult to visualize, ray-intersection can be adapted to operate within it. Parker derived the necessary equations for intersecting a ray with a hexahedron [12]; we essentially used his derivation in order to implement a modified photon mapper in  $\mathbb{R}^4$ . If the 3D domain happened to be organized in another format, such as a tetrahedral mesh, the corresponding intersection calculation would be used.

Illuminating the graph of  $h_3$  with flattened light transport permits sampling of the radiance to be decoupled from sampling of the volumetric grid of the isosurfaces. Figure 6b illustrates this observation. In the left-hand image,  $10^6$  radiance samples are computed on planar slices of the graph. When a level set is illuminated, the net result is merely a collection of radiance samples that may happen to lie on a surface within the volumetric graph. But the graph could be foliated

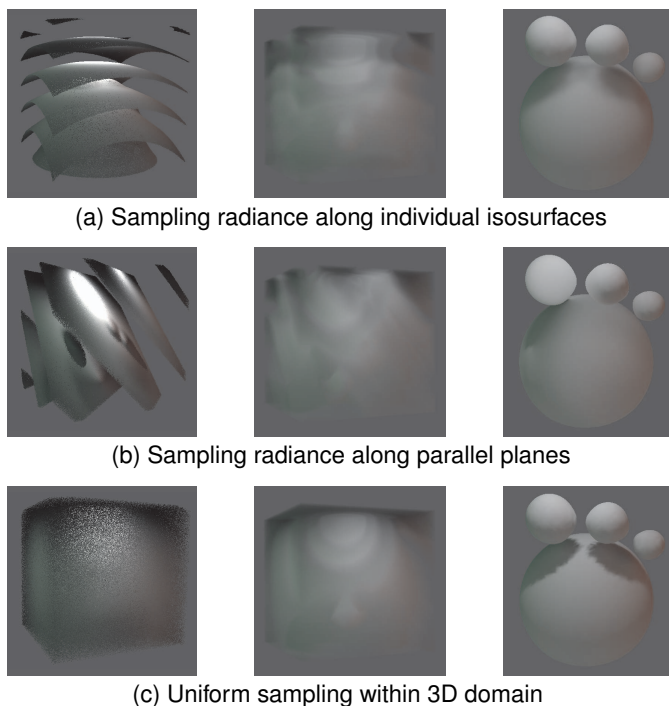


Fig. 6.  $10^6$  radiance samples of function  $h_3$ . Each row shows the pattern of the radiance samples (left), the resulting 3D interpolated illumination grid (middle), and the grid applied as a texture map to an actual isosurface of  $h_3$ . (a) Samples distributed across isosurfaces of  $h_3$  yield visible bands in the illumination grid and overly soft shadows on the large blob when the texture is applied. (b) Samples distributed across diagonal planes in the domain of  $h_3$  yield visible bands as well when interpolated onto the illumination grid. The gaps between sample planes leave shadowed portions of the blobs undersampled. (c) Uniform sampling yields a more accurate illumination grid.

in many different ways, perhaps by parallel diagonal sheets as shown in the image. When the radiance samples are interpolated onto a 3D grid (middle image), banding artifacts are evident along the direction of the sheets. The rightmost image shows an isosurface of  $h_3$  whose illumination comes from this 3D texture. Evidently the top left blob lies between sheets of radiance samples, since it lacks shadowing and reflection from the red and green walls. The correct illumination would have been sampled had the sheets been more closely spaced. It is desirable to distribute the radiance samples in a more optimal way than merely along level sets or other parallel planes in  $\mathbb{R}^4$ .

In Figure 6c, the  $10^6$  samples are distributed uniformly. The left-hand image shows the volume of samples projected down to the 3D domain of  $h_3$ , just as Figures 5a-d shows an overhead view of the surface projected down to the 2D domain of  $h_2$ . When the uniform samples in Figure 6c (left image), are interpolated onto a 3D texture (middle image), the result avoids the sampling artifacts visible in the interpolated textures above it (Figures 6a-b). The only bands in the new 3D texture are regions of isoradiance that actually exist when level sets sweep through the scene. The image on the right shows an isosurface with texture-mapped global illumination using this illumination grid.

### 3 RESULTS AND DISCUSSION

We tested pre-computed illumination using flattened 4D light transport against rendering isosurfaces using photon mapping. Figure 7 illustrates the use of two illumination grids for isosurfaces of the density of protons (red) and neutrons (white) in a computational simulation of the arrangement of nucleons in a neutron star. The nucleon dataset was calculated by Jorge Piekarewicz, Department of Physics at Florida State University; the convolution of the particles with a point spread function to produce continuous 3D density functions was per-

formed by Brad Futch in our lab. It is predicted by simulation that protons and neutrons form clusters at subnuclear densities of around  $10^{14}$  g/cm<sup>3</sup> [9]. The isodensity surfaces for the clusters of protons and neutrons have complicated spatial structure, and thus are candidates for global illumination to improve the user's 3D perception of their shape. Figure 7a shows the scene rendered with local illumination using OpenGL. Figures 7b-c show the scene rendered with global illumination. In Figure 7b, flattened 4D light transport was used to illuminate points on the volumetric graphs of the proton density and of the neutron density in  $\mathbb{R}^4$ . The resulting radiance samples were interpolated onto a 3D illumination grid and applied as textures to the isodensity surfaces. Figure 7c is a reference image in which photon mapping was applied directly to the isodensity surfaces in  $\mathbb{R}^3$ .

The nucleon dataset consists of two components: the proton density and the neutron density, each a uniform grid of resolution  $101 \times 101 \times 101$ . Rendering them simultaneously required generating two textures, one for each density, then extracting and texturing an isosurface from each convolution. The radiance values on the height field were calculated with direct lighting using ray tracing and indirect lighting using the photon map containing about 15 million photons, then were projected and interpolated into a 3D texture. Generating and texturing the level set for each isovalue required an average of 0.885 seconds; after level-set generation, the globally illuminated level set could be rotated in real time (60 Hz) using OpenGL. By comparison, the reference photon-mapped image took 2155 seconds to render using the CPU.

We measured the error between texture-mapped global illumination of isosurfaces using flattened 4D light transport and global illumination of the isosurface itself. For a single vertex, the squared error between the reference surface and the textured surface was calculated as  $|(r_t, g_t, b_t) - (r, g, b)|$  where  $(r_t, g_t, b_t)$  represents the RGB colors of the texture-mapped vertex, each normalized to the interval  $[0, 1]$ , and  $(r, g, b)$  represents the RGB colors for the vertex in the reference surface. The percent RMS error was computed as

$$100\sqrt{\langle \text{Squared vertex error} \rangle}$$

The RMS error for the vertex colors between the texture-mapped isosurface and the photon-mapped isosurface in Figures 7b-c is 5.61%. There are two sources of error: 1) the rendering approximation of estimating indirect illumination directly from the photon map without a final gather, and 2) interpolating the radiance onto a texture and then texture-mapping the radiance onto the level set.

Interpolating the radiance onto a uniform texture reveals sensitivity to the radiance sampling strategy. We compared the accuracy of three different radiance-sampling strategies head to head. These sampling strategies are described below, and are illustrated in Figure 6.

In strategy 1 (**level set**), the radiance is sampled strictly along level sets. Strategy 1 could be accomplished, for example, by generating several level sets and illuminating each one with a 3D renderer. Intuitively, the photons congregate on the level sets, leaving voids in between.

In strategy 2 (**planes**), the radiance is sampled strictly along oblique parallel planes in the volume; this is a straw-man strategy, chosen not because of any expectation that it yields an accurate 3D illumination texture, but because it illustrates that the surfaces on which photons congregate can be completely unrelated to the isosurfaces that are ultimately displayed. Note that each sample captures the radiance of the level set passing through the point, not the radiance resulting from illuminating the plane itself. The situation may be more easily understood by imagining an (admittedly inefficient) algorithm for producing the samples: for each isosurface, flood the scene with photons to determine the radiance at each point on the isosurface, then discard all of the radiance samples except those that happen to lie on (or near) the sampling planes. The result of such a process would be a collection of radiance samples congregating along the oblique planes, even though the samples had been first computed in the context of one isosurface at a time.

In strategy 3 (**uniform**), the radiance samples are uniformly scattered throughout the volume. The situation can be understood by

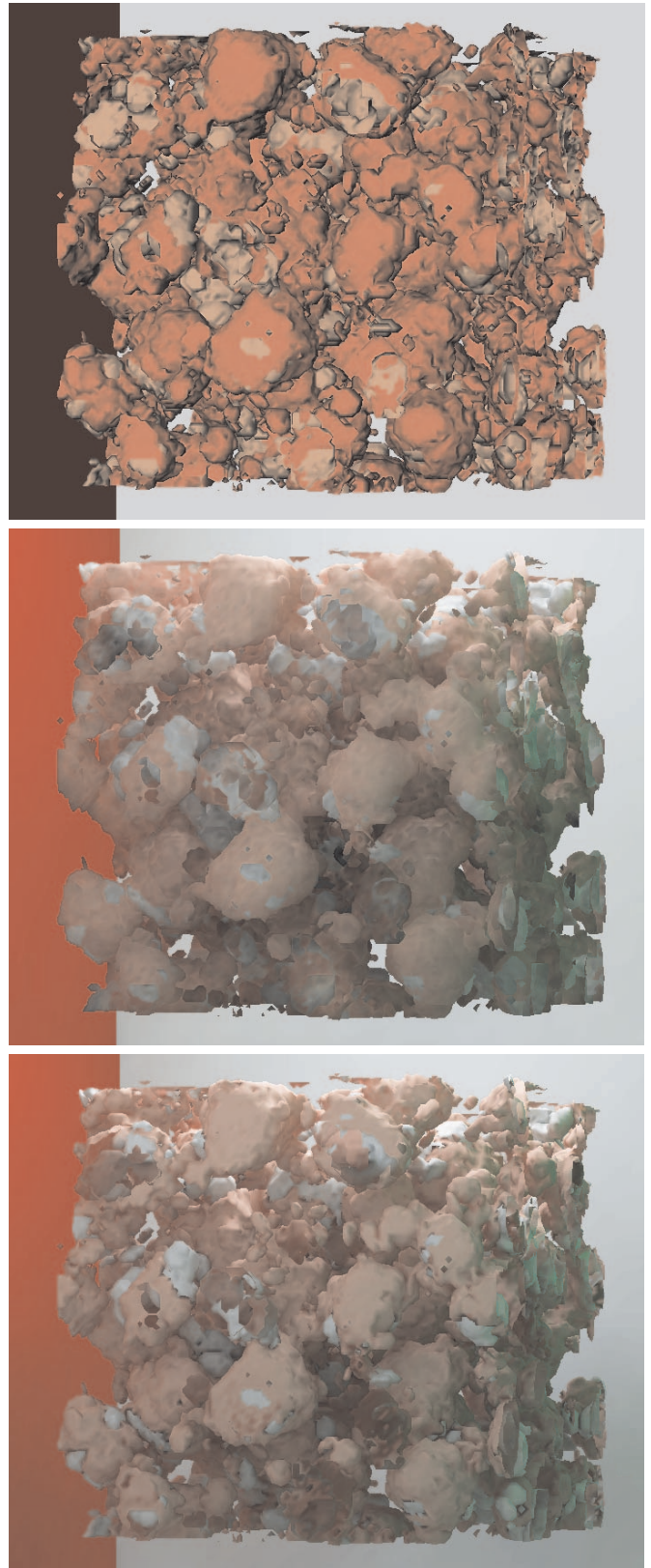


Fig. 7. Isosurfaces of proton density and neutron density in a computational simulation of stellar crust. Top (a): Scene rendered with local illumination (OpenGL). Middle (b): Physically-based illumination applied via texture map from pre-computed illumination grid using on flattened light transport. Bottom (c): Physically-based illumination applied via photon mapping to actual isosurfaces.

imagining an even more inefficient algorithm: for each radiance sample, generate an isosurface through that point; flood the scene with photons to illuminate the isosurface; then discard *all* the photons except the one at the original sample point. If one only has a 3D model available for physically-based illumination, then it is problematic to decouple the radiance-sampling strategy level-set generation. As a result, the radiance samples leave voids between the level sets: where there is no surface, there is no method for calculating the radiance. In order to fill the voids in the radiance samples, one might generate many more level sets, thereby oversampling radiance in regions of the volume where the gradient of the scalar function is large and hence the level sets are packed close together. Such behavior can be seen, for example, in Figure 5a, where level sets (isolines) along the left bump are packed closely together, while level sets near the luminaire (bottom of image) are far apart, leaving voids in between.

By considering the level sets as waterlines of a height field, it becomes possible to sample the radiance of the height field itself, according to a pattern unrelated to the shape of the level sets that foliate it. This is the actual approach we followed with the three sampling strategies, not the pedagogic algorithms (with isosurfaces temporarily generated and then many of their radiance samples discarded) presented above. The error in the sampling strategies is summarized in the table below. The calculations correspond to the scalar function shown in Figure 6. The error is averaged over a level set compared to a reference solution for the level set.

Strategy	samples= $10^6$	samples= $10^7$
	% err	% err
1. Level Set	3.59	3.47
2. Planes	9.55	9.57
3. Uniform	1.29	1.08

The error in strategy 1 is smallest at values corresponding to level sets that were used for sampling the radiance, and largest midway between them; the error in strategy 3 is fairly insensitive to the choice of isovalue. These results quantify the difference that is evident qualitatively in the 2D example of Figure 5. To compare worst-case error, we chose an isovalue (47.0) that lies midway between those (39.0 and 56.7) of two level sets used by strategy 1. The error was found to be lower for the uniform sampling strategy than for the level-set sampling strategy. The 4D lighting model permits the radiance samples to be decoupled from the generation of the level sets, which in turn allows the overall error of the pre-computed illumination texture to be reduced for a fixed number of radiance samples.

It can also be seen that when the number of photons increased, the average error in uniform sampling decreased (improved) more than in parallel-plane sampling or level-set sampling. This behavior can be understood by recognizing that additional photons fill the voids in uniform sampling, but the voids between level sets remain empty when additional photons are introduced using strategies 1 and 2, which require the photons to adhere to level sets or planes. One could insert additional level sets or planes for precomputing radiance values in strategies 1 and 2, but when the total number of photons is held constant, increasing the number of planes or level sets reduces the number of photons sampled on each. In the limit, there would be many level sets (or planes), each having a single photon. This limiting behavior demonstrates the need to rethink the idea of sequentially generating closely spaced surfaces in a volume and individually illuminating them with a 3D lighting model. Once the dense sequence of level sets is viewed as a continuum (the heightfield in  $\mathbb{R}^4$ ) and the lighting model adjusted, the radiance samples need not be constrained to the level sets at all.

We calculated the RMS error for renderings of additional scalar datasets, including a 3D molecular dynamics simulation of a nanofluid, (Figure 8a), 3D magnetic resonance imaging (MRI) of the human brain, (Figure 8b), and 3D confocal microscopy of a mouse neuron (Figure 8c). In these other test cases the RMS error ranged from between 5.7% and 8.1% for texture-mapped illumination and photon mapping of the actual isosurfaces. The grid resolution, number

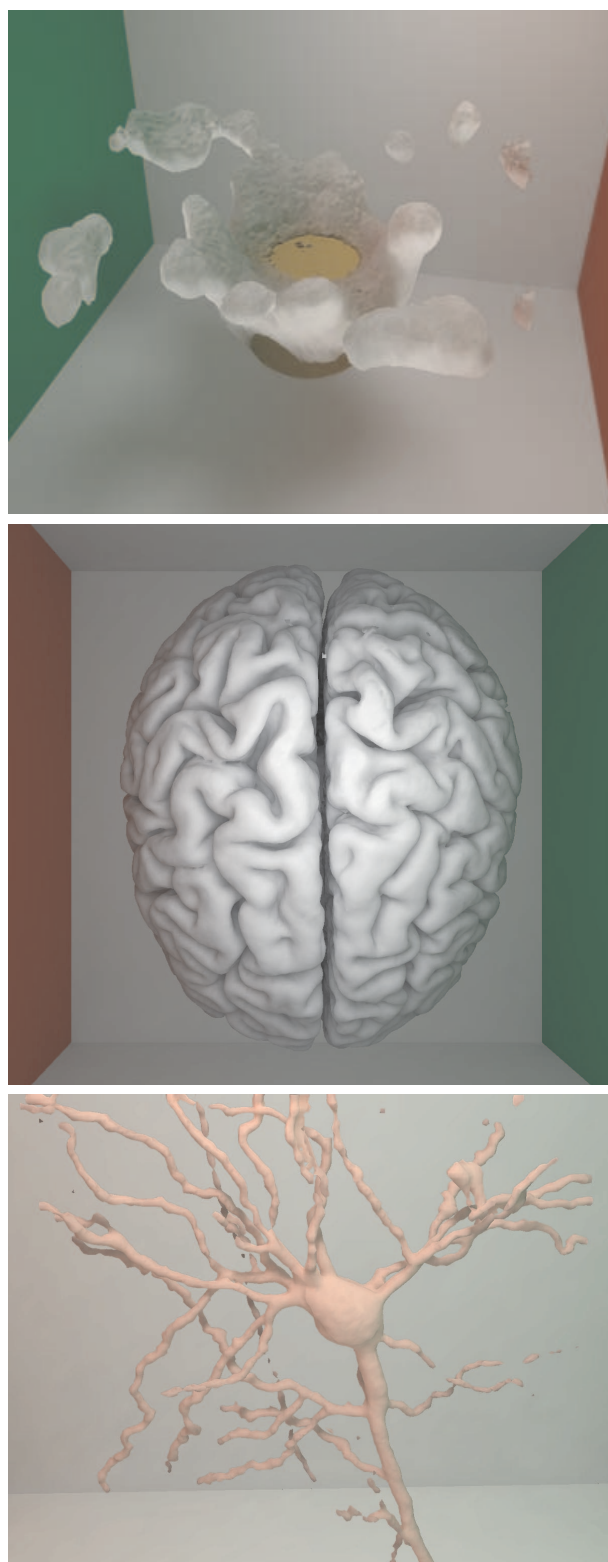


Fig. 8. Example scalar datasets from different domains and displaying different characteristic shapes and topologies. Top (a): Nanofluid simulation with multiple distinct connected components. Data courtesy of M. Yousuff Hussaini, Kayne Smith, and Simon Serge Sablin, Department of Mathematics, Florida State University. Middle (b): Cortex of the brain with multiple folds. Data courtesy of Colin Holmes, McConnell Brain Imaging Center at the Montreal Neurological Institute, McGill University. Bottom (c): Mouse neuron. Data courtesy of Charles Quimet and Karen Dietz, Department of Biomedical Sciences, Florida State University.

of 4D photon samples, time (in seconds) to sample the radiance, and error for these datasets are summarized in the table below. Additional details about timings and errors can be found in [3].

Dataset	Grid	Photons	4D illum (s)	% err
Nucleon	101 <sup>3</sup>	11,000,000	9,685	5.61
Fluid	120 <sup>3</sup>	15,200,000	5,565	8.06
Brain	217 <sup>3</sup>	7,600,000	5,452	6.53
Neuron	150 <sup>3</sup>	11,700,000	4,587	5.70

Although our code used multiple threads to compute the photon maps, the focus of our attention was not on the speed or parallelization of the precomputation phase. Rather, the focus was on demonstrating that a 4D illumination model can be developed for precomputing illumination of level sets, and that such a model permits radiance sampling to be decoupled from level-set sampling. Important questions of both theoretical and practical interest for future research include how to optimally distribute radiance samples to ensure that a given error constraint is met, and how to parallelize the precomputation efficiently.

This paper proposes a model of flattened light transport that operates within  $n$ -dimensional hyperplanes of  $\mathbb{R}^{n+1}$ . The purpose of the model is to permit a luminaire to shine horizontally onto the graph of a scalar function in such a way that light does not leak between different horizontal layers. Since each layer captures a waterline along the graph, the flattened lighting effectively samples the radiance that would have been computed for the corresponding level set down in the domain, one dimension lower. The model can, in principle, be applied to higher-dimensional scalar functions such as time-varying density or pressure.

Flattened light transport allows points on the luminaire to be randomly sampled and points on the graph to be randomly sampled. As a result, sampling of the radiance on the graph is decoupled from constructing level sets, thereby allowing a fixed number of samples to be distributed throughout the 3D domain of a scalar function according to any desired criterion, rather than being distributed strictly on level sets. In particular, the voids between level sets can easily be sampled using the flattened 4D model in order to generate an illumination grid that does not exhibit stairstep interpolation artifacts. The illumination grid can be used as a texture map to illuminate isosurfaces on the fly as they are produced by an isosurface engine in a 3D visualization system.

Illumination grids provide a mechanism for the data analysis workflow to incorporate physically-based illumination on current and legacy visualization systems without requiring any modification to the internal rendering engine or any change to the graphics hardware. The technique described in this paper enhances illumination-grid generation by decoupling the sampling of the radiance from the sampling of the level sets during grid generation. Consequently, radiance samples can be distributed uniformly through a 3D volume rather than being confined to level sets, thereby improving the sampling characteristics of the pre-computation step. We have demonstrated the feasibility of this proposed workflow by incorporating a pre-computed illumination grid as a 3D texture within a commercial visualization software product: without any change to the rendering engine, we demonstrated real-time sweeps through isosurfaces that we textured with pre-computed physically-based illumination.

The authors gratefully acknowledge the improvements suggested by the reviewers. Support for this work was provided by the University of Tennessee, the Science Alliance, the ORNL Scientific Computing group, the National Center for Computational Sciences, the UT/ORNL Joint Institute for Computational Sciences, the Neuroimage Analysis Center (NIH P41RR13218), the National Institute for Computational Sciences (NSF OCI-0711134), and the Biomedical Analysis and Simulation Supercomputer (NIH 1S10RR023069-01). Peter Shirley, Steven Parker, and Henrik Wann Jensen provided helpful conversations and insights. Alan Evans and the Brain Imaging Centre provided office space and access to equipment and data while Banks was in residence at the Montreal Neurological Institute; Carl-Fredrik

Westin and Ron Kikinis provided office space and access to equipment and data while Banks was in residence, and Beason was visitor, at the Laboratory for Mathematics in Imaging, Surgical Planning Laboratory, Department of Radiology, Harvard University. Brad Futch and Hui Song provided valuable software support, physical model testing, and data analysis. Additional support for travel and equipment was provided by Florida State University and the Cornerstone Program.

## REFERENCES

- [1] D. C. Banks. Illumination in diverse codimensions. In *SIGGRAPH 1994: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 327–334, 1994.
- [2] D. C. Banks and K. Beason. Fast global illumination for visualizing isosurface with a 3d illumination grid. *Computing in Science and Engineering*, pages 48–54, Jan/Feb 2007.
- [3] K. M. Beason. Precomputed global illumination of isosurfaces. Master's thesis, Department of Computer Science, Florida State University <http://etd.lib.fsu.edu/theses/available/etd-08152005-222234/>, 26 July 2005.
- [4] K. M. Beason, J. Grant, D. C. Banks, B. Futch, and M. Y. Hussaini. Pre-computed illumination for isosurfaces. In R. F. E. *et al.*, editor, *IS&T/SPIE International Symposium on Electronic Imaging, Conference on Visualization and Data Analysis (E110)*, volume 6060, pages 6060B:1–11, 15–19 January 2006.
- [5] G. P. (editor). *Journal of Physics: Conference Series (Proceedings of SciDAC 2008, July 13–17, 2008, Seattle, WA, USA)*, 125, 2008.
- [6] A. J. Hanson, T. Munzner, and G. Francis. Interactive methods for visualizable geometry. *Computer*, 27(7):73–83, 1994.
- [7] P. S. Heckbert. Radiosity in flatland. *Computer Graphics Forum (EUROGRAPHICS '92 Proceedings)*, 11(3):181–192, 1992.
- [8] S. R. Hollasch. Four-space visualization of 4D objects. Master's thesis, Department of Computer Science, Arizona State University, 1991.
- [9] C. Horowitz, M. Perez-Garcia, D. Berry, and J. Piekarewicz. Dynamical response of the nuclear “pasta” in neutron star crusts. *Phys Rev C*, 72:031001, 2005.
- [10] H. H. Hu, A. A. Gooch, W. B. Thompson, B. E. Smits, J. J. Rieser, and P. Shirley. Visual cues for imminent object contact in realistic virtual environments. In *Proceedings of Visualization 2000*.
- [11] H. W. Jensen. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., 2001.
- [12] S. Parker, M. Parker, Y. Livnat, P.-P. Sloan, C. Hansen, and P. Shirley. Interactive ray tracing for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):238–250, 1999.
- [13] L. Randall. *Warped Passages: Unraveling the Mysteries of the Universe's Hidden Dimensions*. Ecco, 2005.
- [14] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, New York, NY, USA, 1968. ACM Press.
- [15] D. Stalling, M. Westerhoff, and H.-C. Hege. amira: A highly interactive system for visual data analysis. In C. D. Hansen and C. R. Johnson, editors, *The Visualization Handbook*, chapter 38, pages 749–767. Elsevier, 2005.
- [16] A. J. Stewart. Vicinity shading for enhanced perception of volumetric data. In *IEEE Visualization*, pages 355–362, 2003.
- [17] C. Weigle and D. C. Banks. A comparison of the perceptual benefits of linear perspective and physically-based illumination for display of dense 3d streamtubes. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1723–1730, 2008.
- [18] C. Wyman, S. Parker, P. Shirley, and C. Hansen. Interactive display of isosurfaces with global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):186–196, March/April 2006.