Paper:

Spencer, B., Jones, M. & Lim, I. (2015). A Visualization Tool Used to Develop New Photon Mapping Techniques. *Computer Graphics Forum, 34*(1), 127-140.
http://dx.doi.org/10.1111/cgf.12464

A Visualisation Tool used to Develop New Photon Mapping Techniques

B. Spencer[1], M. W. Jones[1] and I. S. Lim[2]

[1] Visual and Interactive Computing Group, Swansea University, UK
[2] Bangor University, UK

Abstract

We present a visualisation tool aimed specifically at the development and optimisation of photon map denoising methods. Our tool allows the rapid testing of hypotheses and algorithms through the use of parallel coordinates, domain-specific scripting, color mapping and point plots. Interaction is carried out by brushing, adjusting parameters and focus-plus-context, and yields interactive visual feedback and debugging information. We demonstrate the use of the tool to explore high-dimensional photon map data, facilitating the discovery of novel parameter spaces which can be used to dissociate complex caustic illumination. We then show how these new parameterisations may be used to improve upon pre-existing noise removal methods in the context of the photon relaxation framework.

1. Introduction

Computer graphics research has focused on the development of ever more complete and efficient solutions to the global illumination problem [Kaj86]. Aided by an exponential increase in storage and processing power, the state of the art has now advanced to the point where true photorealism is attainable within practical timeframes. This has been assisted by exploiting coherence and computational redundancy and has yielded dramatic improvements in both quality and performance while still treading the fine line of perceptual error.

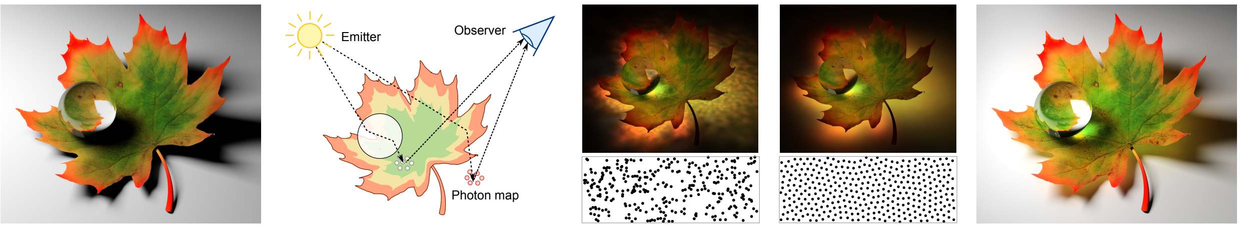
Of the many notable contributions that have had a lingering impact, the photon map [Jen96] presents a conceptually simple, extensible and powerful global illumination solution. Its core paradigm exploits spatial coherence in surface irradiance, bridging and combining eye and light paths in the spatial domain. What makes photon mapping particularly powerful is its strategy of caching radiant flux as a geometrically decoupled point cloud. This approach has been praised for its efficiency thanks to a conservative, sparse representation of flux data with locally adaptive properties. Furthermore, its versatility has promoted rapid adaptation to caustics, diffuse interreflection and participating media.

One of the major disadvantages of photon mapping is its susceptibility to low-frequency image noise caused by point discrepancy and flux variance in its underlying dataset. Numerous strategies have been proposed to address this problem, many of which focus on filtering out the noise by modifying the kernel used in the radiance reconstruction. A more recent development and a relative departure from this convention is photon relaxation [SJ09] which applies iterative point redistribution to diffuse away noise prior to rendering.

Among the more prominent benefits of photon relaxation is its relatively high resilience to bias [SJ13a]. Although this property has also been the goal of numerous other noise reduction algorithms, the unique way in which the algorithm redistributes individual photons allows not only for structure adaptation, but also for structure *dissociation*. In the context of the photon map, ancillary data encoded within each photon's ray path aids in structural separability when parameterised within a high-dimensional domain.

The task of finding a parameterisation that guarantees structure dissociation is non-trivial because the dimensionality of the ray path data must first be reduced before it can be integrated into the photon kd-tree. Crucially, the problem is not well suited to regression or other methods based on error minimisation because of the extremely large search spaces involved.

In this paper we describe the problem in greater detail and suggest that the most efficient means of designing an effective parameterisation is with close user scrutiny and interaction. We then introduce a toolkit designed specifically for



(a) (b) (c) (d) (e)

Figure 1: *Photon mapping and relaxation. (a) A model of a maple leaf rendered with only direct (single-bounce) illumination. (b) Photons traced from the emitter are stored and later used to reconstruct more complex components of illumination - in this case, caustics and sub-surface scattering. (c) When isolated, reconstructed illumination from the photon map shows a high degree of noise. (d) After photon relaxation, noise is substantially reduced. (e) The final composite image.*

the visualisation of photon data. The principal contributions are:

- An interface for intuitive exploration of multi-dimensional photon data using point projections, color maps and parallel coordinates (including the introduction of elliptical axes and density estimation as a visual cue within parallel coordinates).
- A domain-specific language (DSL) allowing candidate parameterisations to be rapidly and flexibly prototyped.
- A loupe tool for detailed examination of localised neighbourhoods within the photon distribution.

We demonstrate a number of novel, practical applications, including the construction of a more robust parameter space for the dissociation of overlapping caustics, and a method for automatically constraining the search radius in parameter space.

2. Related Work

2.1. Parameterised Noise Filtering

Filtering noise from an input signal is a fundamental operation employed throughout the field of computer rendering. While many algorithms use naïve filters or those which adapt solely to changes across the image domain, an increasing number take advantage of the extra information made available by the ray tracer to improve filtering performance.

Dammertz et al. [DSHL10] use surface normal and world space positional data to create an edge-stopping function for the À-Trous wavelet filter. Lehtinen et al. [LAC*11] exploit anisotropy and temporal coherence to upsample a 5D light field resulting in an improvement in quality while avoiding smoothing artifacts. More recently, Sen and Darabi [SD12] compute the functional relationship between sample values using an array of feature vectors to isolate noise generated from random parameter sampling.

Filtering guided by data from the ray tracer has also been used to improve the performance of the photon map. Using an adaptation of Igehy's ray differential framework [Ige99], Schjøth et al. [SFES07] show significant improvements in the fidelity of specular caustics by shaping the kernel filter of the radiance estimation based on the footprints of captured photons. Chen et al. [CTC10] proposed a method of reducing bias by grouping photons with coherent paths and estimating the extant radiance over the convex hull of the resulting clusters. Havran et al. [HBHS05] proposed a solution to the problems of boundary and topology bias by storing photons as rays instead of particles. This approach has the benefit of including the contributions of photons that pass close to, but do not intersect a particular surface.

2.2. Data Visualisation

Systems such as Voreen [MSRMH09] and that of Rieder et al. [RPLH11] present a rapid prototyping approach for volume rendering where modular shader blocks can be combined to create a complete pipeline. The system relies on establishing a signature for the blocks that allows combination and communication to take place. Custom code can be inserted into modules. Recent work by Etiene et al. [EJR*13] demonstrate errors in Voreen (and VTK) found using a verification process. Our work allows a customised rapid prototyping for exploratory analysis of photon mapping and endeavours to remove errors through visualisation. Verification is a step that would be taken once the principles and suitability of a method are firmly understood and established. Laramee [Lar10] issues a set of guidelines for how to debug software using visualisation.

Gribble et al. [GFE*12] create rtVTK as a visualisation support system for ray based renderers. Like us, they expound the view that realising a visualisation of the ray tracer will lead to a deeper understanding of the process, further leading to new insights. They also highlight the use of the system for quickly identifying and therefore removing a previously undiscovered (and probably undiscoverable) bug.

Diderot [CKR*12] is a domain specific language targeted at tensor field visualisation (primarily biomedical). The aim is to provide a higher level language for developing visualisation codes, but still maintaining performance and scalability. We provide a DSL for photon mapping, but augment it with visualisation tools for algorithm insight and with a secondary effect of aiding debugging.

Large data sets can lead to occluded (over-rendered) and cluttered displays of the data and difficulty determining relationships between non-adjacent axes. Geng et al. [GPL*11] presented one solution to overcome the large data problem by employing a histogram oriented in the mean outgoing direction of the axis. Illustrative techniques have been applied to rendering [MM08], rendering data as a continuous density field [HW09] and preserving outliers [NH06]. Data can be clustered prior to rendering, or during visualisation [ZYQ*08]. Claessen and van Wijk [CvW11] give new ways to control the placement of axes in parallel coordinate plots. Qu et al. [QCX*07] introduce an 'S' shaped axis to integrate vector data into PCPs.

We are interested in employing parallel coordinates to discriminate our data, and present a new visualisation that incorporates mixing open and closed axes together with kernel density estimation to provide extra visual cues to the data clustering along the axis. We also employ cubic splines to avoid visual clutter occurring when we draw across the open axis.

2.3. Control and Visualization of Light Transport

Reiner et al. [RKRD12] provide a nice collection of visualization tools for understanding complex light transport within scenes. Like our method, photons are augmented with additional data about their interaction with the scene. In one example, they utilise this data to rasterize photon paths to a volume of data which can be rendered within the scene, thus depicting the flow of light (such as caustics) or shadows. These tools can aid understanding of complex lighting and complement our techniques. Our work differs in that we provide tools and a DSL aimed at researchers for improving post-processing of photon maps, specifically relaxation for noise reduction whilst maintaining fine lighting structure. Schmidt et al. [SNM*13] follow up the work by allowing users to select, visualise and manipulate light transport. Such a tool is aimed at artists looking to achieve certain visual effects within images. Pellacini [Pel10] demonstrates envyLight, a system for separating illumination contributions (from environment maps) into separate layers that are subsequently edited and recombined, thus giving artistic control over illumination. Nowrouzezahrai et al. [NJS*11] also address artistic control of lighting by introducing physically based modelling for control and procedural shading. Our tool also offers end-users insight into light transport, and allows precise control and technical rendering of scenes compared to the coarser intuitive controls provided by the above approaches. Thus, it could complement those artistic interfaces. Progressive photon mapping [HOJ08] is particularly successful at rendering caustic illumination by utilising progressively shrinking kernel bandwidth during density estimation. Our approach produces view independent photon maps. Progressive photon relaxation [SJ13b] combines relaxation with a progressive approach to produce view independent photon maps, but requires a lengthy computation time of similar time to Progressive photon mapping.

3. Problem Background

In this section we discuss the photon relaxation algorithm in greater depth, highlight its major shortcomings, discuss the source of these problems, as well as the techniques that can be employed to address them.

3.1. Photon Relaxation and Migration Constraints

In Figure 1(a), a glass marble rests on a maple leaf. Light passing through both objects is cached by the photon map and is later reconstructed by the rendering pass (Figure 1(b)). Isolating this contribution in Figure 1(c) reveals characteristic, blotchy artifacts due to photon discrepancy and flux variance. A schematic of the distribution offers further insight by showing the stochastic placement of photons which result in an inconsistent estimate of incident flux per unit area. By iteratively repelling each photon away from its neighbors, the distribution is relaxed, resulting in a much more consistent estimate of area and a corresponding reduction in image noise (Figure 1(d-e)).

Though effective, point relaxation also has the effect of degrading salient visual features of the distribution. These features include discontinuous transitions in illumination such as shadow boundaries, as well as subtle, interlaced or overlapping structure. An example is presented in Figure 2 where a plastic thumb tack is lit from above by a bright point light. If we relax the distribution by naïvely seeking to achieve point equidistribution, noise is completely eliminated but at the expense of the fine detail which makes the caustic so attractive (Figures 2(a) and 2(b)).

The deleterious effects of relaxation can be controlled by inhibiting migration in regions where diffusion would otherwise result in an increase in bias. This goal is achieved using a series of constraints which are applied to each photon before the relaxation pass. By this process, smoothing becomes locally adaptive and noise can be removed while still preserving the general structure of the photon distribution. The result of photon constraints are shown in Figure 2(c). Note how the original shape of the caustic is almost completely preserved while still ensuring noise is diffused away.

3.2. Gradient Ambiguity due to Estimator Bias

The point repulsion algorithm as defined by Spencer and Jones [SJ09] induces photons to migrate along the axis of the flux density gradient, ∇B, at a rate proportional to its magnitude. The derivative of flux density ranges from zero

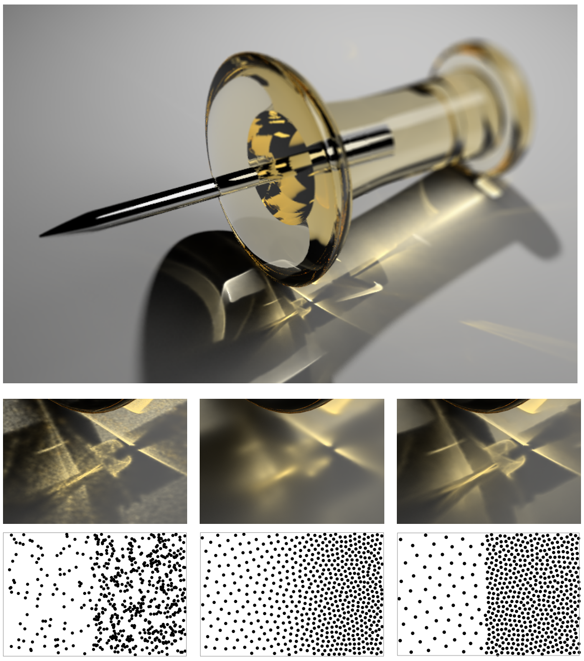

(a) Unrelaxed (b) Unconstrained (c) Constrained

Figure 2: *The importance of photon constraints. (a) A close-up of the caustic reveals noise caused by discrepancy in the distribution. (b) Naïve relaxation of the distribution removes noise but degrades detail. (c) Limiting the motion of photons along the flux density gradient preserves detail while allowing noise to diffuse away.*

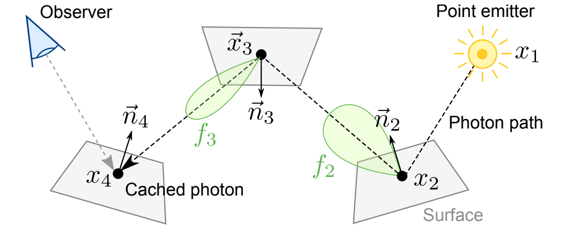


Figure 3: *Photon path space. Node $\vec{x}_1$ corresponds to the photon's origin, $\vec{x}_2$ and $\vec{x}_3$ are points of interaction with reflective geometry and $\vec{x}_4$ the spatial position of the stored photon. $\vec{n}_2 ... \vec{n}_4$ are the surface normal vectors and f_2, f_3 are the surface BRDFs.*

where the function does not change, to infinitely large where the function becomes discontinuous. Ideally, photons on an infinite gradient should be maximally constrained since any movement across it would visibly degrade the transition.

Problems arise because the derivative of a discontinuity is a delta function with a value of plus or minus infinity over an infinitesimal interval. Accurately estimating this derivative from the photon map is impossible because bias in the kernel effectively smooths the delta function in the spatial domain and attenuates at its peak, making it finite and introducing an unresolvable ambiguity.[†]

In general terms, the magnitude of the constraint computed by a biased gradient estimator correlates to the relative visibility of a given structure within the caustic. We refer to Figure 7(g) for an example. Here, the strong discontinuity on the edge of the caustic is very distinctive and has been correspondingly well constrained. Conversely, the secondary arc visible on the interior of the caustic is harder to discern and hence is more weakly constrained. In order to prevent loss of detail, photons on *both* boundaries should be unable to move so as to prevent adjacent regions from bleeding together.

Our research into addressing photon constraints forms the focus of this paper. Since the gradient estimator from which photon constraints are derived is susceptible to bias, we investigate means of decomposing caustic illumination so that ambiguous estimates can be effectively resolved.

3.3. Dissociating Structure in the Photon Distribution

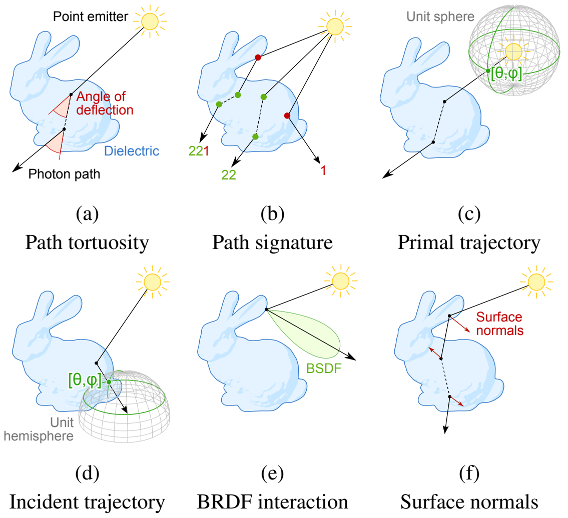
Thus far we have explored how overlapping features within the caustic overwhelm the gradient reconstruction kernel and result in weakened constraints. This result leads us to the question of whether, for a given feature, interference from these extraneous elements can be disregarded so as to yield a cleaner estimate and more robust constraint.

Structure differentiation based on these criteria requires that each filament (i.e., an edge or other visual cue) of the distribution exist in isolation from its fellows. Since the position of each photon in Euclidean world space must remain unchanged (at least initially), a secondary space must be created which allows this separation. We refer to this new domain as *photon parameter space*. Assigning each photon a position within it form the basis of the research in the remainder of this paper.

Here we are at an advantage since, in addition to the position and power of stored photons, we have at our disposal a number of additional attributes which are output by the ray tracer. These include data about the photon ray path, the objects it intersects and the emitter from which it originates (Figure 3). The sum total of these data define a hyperspace within which each photon resides.

In the case of the problem posed in Section 3.2, the question we are looking to answer is how best to discover a projection of the complex multidimensional parameter space of photon data so that structure within the caustic can be both efficiently and effectively differentiated. When working to create a suitable parameter space for the photon map, we seek to answer the following questions:

[†] For a more detailed exposition of this error, we refer readers to Spencer and Jones [SJ13a].



(a) Path tortuosity

(b) Path signature

(c) Primal trajectory

(d) Incident trajectory

(e) BRDF interaction

(f) Surface normals

Figure 4: *Candidate parameterisations for photon paths. (a) Path tortuosity is a scalar measure of how much a ray path is bent as it is reflected and refracted . (b) A path signature is an encoding of a photon's interaction as it is propagated throughout the scene. (c-d) The primal and incident trajectories are vector quantities that define the direction of the photon as it leaves the emitter or arrives on a surface. (e) BSDF interaction indicates to what degree a photon is scattered. (f) Surface normals at ray intersections may be integrated to create more elaborate parameterisations (explored in Section 5.3).*

- Given a specific photon distribution, is it possible to discover a parameterisation that will robustly dissociate all important structures and other visual cues?
- If such a parameterisation exists, how can we seek to minimise the number of dimensions required to encode it?
- Can it be extrapolated within realistic computational and memory bounds?
- Is it possible to generalise the parameterisation to arbitrary distributions?

In order to address these problems is becomes apparent that having a tool to visualise photon map data, interrogate photon statistics, and expand in order to test new hypotheses would be most useful. Only through the use of this visualisation tool (Section 4), combined with the ideas about how to approach the problem, were we able to develop successful techniques for producing high quality photon maps.

3.4. Candidate Parameterisations

To better illustrate the abstract notion of a photon parameterisation, we define a series of scalar, vector and tensor parameter spaces that might conceivably be extrapolated from a ray path (Figure 4).

3.4.1. Path Tortuosity

The tortuosity of a path refers to the degree with which is twists and turns as it is reflected, refracted or scattered by intersecting geometry or participating media (Figure 4(a)), and is inspired by our work on biological problems [GJL*09] and streamline clustering [MJL*13]. Since a path is comprised of a discrete number of elements (rays), we propose defining the tortuosity of a photon path as the mean angle of deflection at each intersection, normalised to the maximum possible deflection:

$$\gamma(\vec{x}_1,\ldots,\vec{x}_m) = \frac{1}{\pi(m-2)}\sum_{i=2}^{m-1}\arccos\left(\frac{(\vec{x}_i-\vec{x}_{x-1})\cdot(\vec{x}_{i+1}-\vec{x}_i)}{\|\vec{x}_i-\vec{x}_{x-1}\|\|\vec{x}_{i+1}-\vec{x}_i\|}\right) \tag{1}$$

where $\vec{x}_1,\ldots,\vec{x}_m$ is the list of path vertices. Note that this equation requires m to be greater than 3, however the photon map is typically configured to ignore paths that have not undergone at least once bounce.

3.4.2. Path Signature

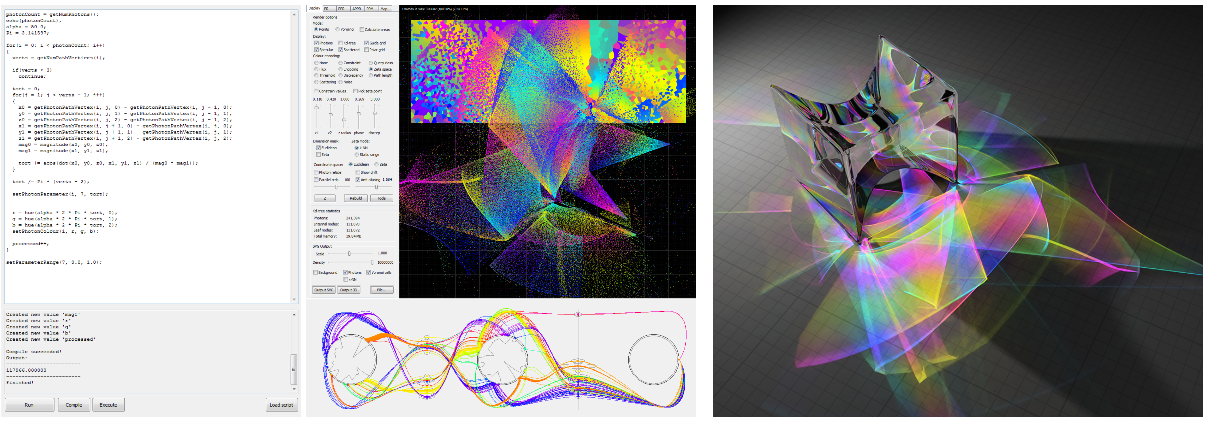
We define the signature of a path to be a discrete scalar encoding derived from both the number and nature of interactions that a path undergoes with intersecting geometry. Each intersection is encoded by a digit position in an integer of radix 3. The least significant digit corresponds to the intersection of the primal ray, the second-least to the intersection of its child and so on, moving toward the most significant digit for the length of the path. A reflective interaction is assigned a value of 1, a transmissive interaction a value of 2. For a given path, $\vec{x}_1,\ldots,\vec{x}_m$, this can be defined as follows:

$$S(\vec{x}_1,\ldots,\vec{x}_m) = \sum_{i=2}^{m-1} 3^{i-2}\cdot\begin{cases} 1 & \textit{if } \vec{x}_i \textit{ is reflected} \\ 2 & \textit{if } \vec{x}_i \textit{ is transmitted}\end{cases} \tag{2}$$

Path signature encoding is illustrated in Figure 4(b). The left-most ray undergoes three bounces: a reflection which is encoded by a 1 in the first digit position, plus two transmissions both of which are encoded by 2s in the second and third digit positions. Hence, the non-terminating vertices result in the encoding 221 (or 25 in base 10) for this path.

3.4.3. Photon Trajectory

A photon's trajectory represents the pair of spherical coordinates which encode the ray's direction upon leaving the emitter or arriving at an intersecting surface. For omnidirectional point lights (as illustrated in Figure 4(c)) this requires 3 degrees of freedom to completely represent the space over the unit sphere. For surfaces and planar emitters (Figure 4(d)), 2 degrees are sufficient since only the unit hemisphere is considered and may be flattened into a plane.



```
photonCount = getNumPhotons();
echo(photonCount);
alpha = 50.0;
Pi = 3.141597;

for(i = 0; i < photonCount; i++)
{
  verts = getNumPathVertices(i);

  if(verts < 3)
    continue;

  tort = 0;
  for(j = 1; j < verts - 1; j++)
  {
    x0 = getPhotonPathVertex(i, j, 0) - getPhotonPathVertex(i, j - 1, 0);
    y0 = getPhotonPathVertex(i, j, 1) - getPhotonPathVertex(i, j - 1, 1);
    z0 = getPhotonPathVertex(i, j, 2) - getPhotonPathVertex(i, j - 1, 2);
    x1 = getPhotonPathVertex(i, j + 1, 0) - getPhotonPathVertex(i, j, 0);
    y1 = getPhotonPathVertex(i, j + 1, 1) - getPhotonPathVertex(i, j, 1);
    z1 = getPhotonPathVertex(i, j + 1, 2) - getPhotonPathVertex(i, j, 2);
    mag0 = magnitude(x0, y0, z0);
    mag1 = magnitude(x1, y1, z1);

    tort += acos(dot(x0, y0, z0, x1, y1, z1) / (mag0 * mag1));
  }

  tort /= Pi * (verts - 2);

  setPhotonParameter(i, 7, tort);


  r = hue(alpha * 2 * Pi * tort, 0);
  g = hue(alpha * 2 * Pi * tort, 1);
  b = hue(alpha * 2 * Pi * tort, 2);
  setPhotonColour(i, r, g, b);

  processed++;
}

setParameterRange(7, 0.0, 1.0);
```

```
Created new value 'mag1'
Created new value 'r'
Created new value 'g'
Created new value 'b'
Created new value 'processed'

Compile succeeded!
Output:
------------------------
117966.000000
------------------------
Finished!
```

Figure 5: *The main interface for our photon map visualisation tool. Left: user-defined scripts can be loaded and edited then compiled and executed on the photon map. Visible in this example is code to compute photon path tortuosity (Equation 1). Middle: the false-color encoding resulting from applying the open script to the photon distribution. Visible in this view is the photon Voronoi tessellation and the parallel coordinates of the extrapolated parameterisation. Right: applying the false-colour encoding to the final render.*

Given the normalised trajectory vector, $\vec{t}$, and an orthonormal basis, $[\vec{u}_n, \vec{v}_n, \vec{w}_n]$, where $\vec{w}_n$ is equal to the path normal, $\vec{n}_i$, we define the spherical coordinates θ and ϕ as:

$$\begin{aligned} \theta &= \arccos(\vec{t} \cdot \vec{w}_n) \\ \phi &= \operatorname{atan2}(\vec{t} \cdot \vec{u}_n, \vec{t} \cdot \vec{v}_n) \end{aligned} \tag{3}$$

4. Exploring Photon Parameterisations

Thus far we have defined the problem of interference when estimating flux gradients and proposed a range of candidate parameter spaces which might help reduce estimate ambiguity. What we require now is a means to test whether or not they are effective at dissociating structure within the distribution. In the following sections we explore a range visual and interactive techniques designed to allow the user to input, verify, deploy and (if necessary) refine novel parameterisations.

4.1. Input via Imperative Scripting

Scripting languages are tools that allow the automation of complex processes in a compact and versatile way. The versatility of imperative scripts makes them an ideal bridge between the mathematical abstraction of a parameter space and the physical environment of a global illumination renderer. Once reinterpreted using the syntax of the language, parameterisations may be effectively extrapolated from photon path space. More importantly, error cases and outliers are effectively handled - a useful tool which can be used to improve or debug ineffective candidates.

The syntax of our scripting language is broadly similar to C. Common photon map operations are exposed by an API which encapsulates getting and setting data about each photon's path and declaring and storing derived data. For example, declaring a new parameterisation is achieved by calling `declarePhotonParameter(dimension, minBound, maxBound)` and later accessed by the function `setPhotonParameter(photon, dimension, value)`.

The script in Algorithm 1 reproduces the operations defined in Equation 1, computing and storing each photon's path tortuosity. The resulting scalar output is also mapped to a colour encoding which is cached at the photon. Prior to execution, photon's spatial positions are defined as the $[x,y,z]$ triple of its spatial position. After the script has successfully terminated, each coordinate is extended to a $[x,y,z,\gamma]$ quad which incorporates the newly derived tortuosity parameter.

4.2. The Photon Exploration Loupe

To better facilitate assessment of new parameterisations, we allow the user to select a photon and automatically query and display data about it and its immediate neighbourhood. We refer to this tool as the exploration loupe since it allows regions of interest to be rapidly highlighted and examined.

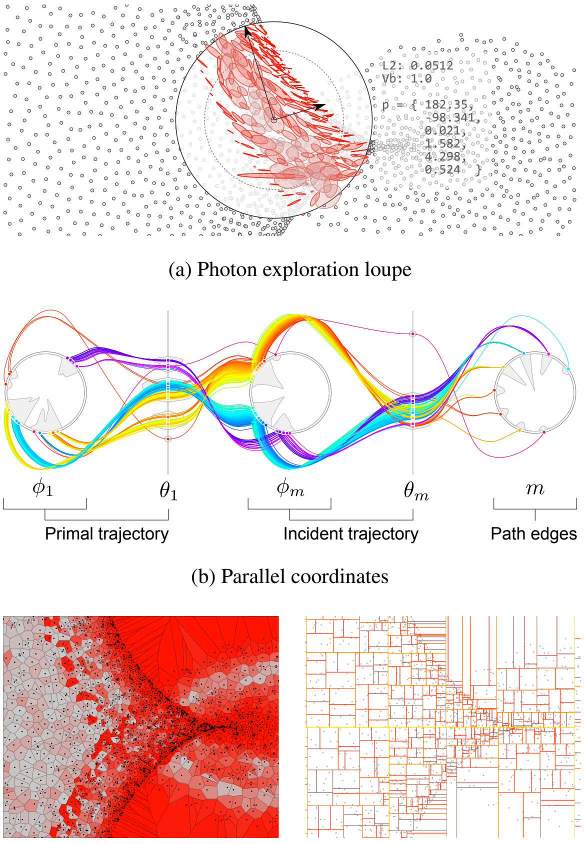
A schematic of the loupe tool is illustrated in Figure 6(a). The underlying distribution (rendered in gray) is a close-up of the cusp of the cardioid caustic from the gold ring scene in Figure 7. The 250-nearest neighbours to the photon at the centre of the loupe have been queried and highlighted, and three sets of attributes have been visualised:

- The derived orthogonal constraining basis, $[\vec{u}_b, \vec{v}_b]$, for the selected photon, visualised as a pair of perpendicular arrows. These indicate by what degree and in what direction migration of this photon will be limited and are calculated using PCA [SJ13a].

Algorithm 1 DEFINEPATHTORTUOSITY()

```
alpha = 50;
expose alpha range 1, 1000
X = 0; Y = 1; Z = 2; GAMMA = 3;

declarePhotonParameter(GAMMA, 0.0, 1.0);

for(i = 0; i < getNumPhotons(); i++)
{
  verts = getNumPathVertices(i);

  if(verts < 3) continue;

  tort = 0;
  for(j = 1; j < verts - 1; j++)
  {
    x0 = getPathVertex(i,j,X) - getPathVertex(i,j-1,X);
    y0 = getPathVertex(i,j,Y) - getPathVertex(i,j-1,Y);
    z0 = getPathVertex(i,j,Z) - getPathVertex(i,j-1,Z);
    x1 = getPathVertex(i,j+1,X) - getPathVertex(i,j,X);
    y1 = getPathVertex(i,j+1,Y) - getPathVertex(i,j,Y);
    z1 = getPathVertex(i,j+1,Z) - getPathVertex(i,j,Z);
    mag0 = magnitude(x0, y0, z0);
    mag1 = magnitude(x1, y1, z1);

    tort += acos(dot(x0, y0, z0, x1, y1, z1)
        / (mag0 * mag1));
  }

  tort /= Pi * (verts - 2);

  setPhotonParameter(i, GAMMA, tort);

  r = hue(alpha * 2 * Pi * tort, 0);
  g = hue(alpha * 2 * Pi * tort, 1);
  b = hue(alpha * 2 * Pi * tort, 2);
  setPhotonColourEncoding(i, r, g, b);
}
```

- The post-relaxation drift of each neighbour. These ellipsoids represent how much each photon has moved within the confines of its constraining basis. The larger the ellipsoid, the greater the migration pressure exerted on the photon by its neighbours. In regions of high constraint, this may indicate weak gradient detection which may further suggest that the parameterisation is not effective.
- Selectable numerical diagnostic information. Visible to the right of the loupe are the second eigenvector of the local PCA decomposition, the degree of constraint of $\vec{v}_b$ and the tuple of spatial coordinates for the photon.



(a) Photon exploration loupe

(b) Parallel coordinates

(c) Voronoi tessellation

(d) Kd-tree visualisation

Figure 6: *Photon map diagnostic tools. (a) The neighborhood exploration loupe. Here, the 250 photons nearest the point in the centre of the reticule are retrieved and highlighted. Structural information is then computed and overlaid together with data about the distribution (spatial coordinates, Eigenvalues, tortuosity). (b) Parallel coordinates for the 250-nearest neighbors in world space. Primal and incident photon trajectories are plotted together with the number of edges in its path. The colour encoding is derived from the first parameter, ϕ_1. (c-d) The Voronoi tessellation and kd-tree partition of the point distribution.*

4.3. False-Colour Encodings

One of the most intuitive means of verifying a parameterisation at-a-glance is by applying a false-colour encoding to each photon based upon its position in parameter space. This lends caustic filaments instant visual separability given that the human eye is acutely sensitive both to variations in brightness as well as intensity and intermixing of colour spectra. Retinal tristimulus means that we can comfortably differentiate colours in a space of up to three dimensions without ambiguity. Theoretically therefore, a three-dimensional parameterisation can be mapped bijectively to the RGB colour space. If the dimensionality of the parameter space is less than three, we can devolve to a more primal colour model which affords a greater degree of distinction.

Figure 5 demonstrates how a colour encoding may be derived from the photon parameterisation and how this maps to the photon distribution in world space. Here, the 2-dimensional primal trajectory space (Figure 4(c)) of the photon map has been mapped to the RGB colour space using the following association:

$$\begin{aligned} r_c &= \tfrac{1}{2} + \tfrac{1}{2}\cos(\theta\cos\phi) \\ g_c &= \tfrac{1}{2} + \tfrac{1}{2}\sin(\theta\sin\phi) \\ b_c &= 1 - rg \end{aligned} \qquad (4)$$

Immediately we see a visual distinction between overlapping structures which indicates that this parameterisation is at least partially effective.

Problems with this approach arise when only small regions of the photon distribution are inspected at any one time (for example, using the loupe tool in Figure 6(a)). In this case, colour variation (dynamic range) between neighbours may be too narrow to accurately gauge whether one filament of the distribution has been adequately separated from another. In this case we find it useful to apply a periodic function that oscillates backward and forward through the colour space many times over the corresponding interval in parameter space. Even using this work-around, it is still possible that two overlapping regions will be assigned the same colour while not sharing the same parameter space coordinates.

This lack of dynamic range is a limitation of colour encoding which makes it unsuitable for more in-depth analysis. Furthermore, stratification of photons between object and parameter spaces, together with potential correspondences between other, non-encoded spaces are similarly not communicated. We utilise parallel coordinates to solve this problem.

4.4. Parallel Coordinates

Parallel coordinates [Ins85] is a ubiquitous technique for rendering high-dimensional, multivariate data as a simple visualisation. Points are plotted by projecting their coordinate elements onto a series of parallel axes and linking them with a line or higher-order curve. The technique is beneficial because interaction with the data can lead to insight, particularly when including expert knowledge in developing domain rules for discriminating data, and the capability to show additional derived data channels at run-time [TFA*11].

To overcome the limitations of false-colour encodings, we employ parallel coordinates to assist in parameter space verification. Rather than attempting to represent the coordinates of every photon on the same graph, we use the exploration loupe (Section 4.2) to isolate the group of k neighbours nearest to the cursor. A configurable tuple of parameters from each member of k is plotted and linked by a curve which is then tinted according to the chosen colour encoding (Section 4.3). The plot updates as the loupe is moved across the distribution thereby allowing the user to scan for clusters which correspond to separable features.

A parameterisation may exist either in an open Euclidean space or in a closed elliptic or other manifold space. Hence, we use a dual notation whereby each axis is rendered as either a straight line between limits, or over an ellipse (Figures 6(b) and 7(h)). Elliptic projections are especially versatile because of their intrinsic periodicity. This property means that, in exchange for sacrificing the bijective mapping between axis and parameter, we can artificially increase point separation to enlarge clusters while still preserving local continuity. Since certain parameterisations result in many photons sharing the same parameter (for example, path signature in Equation 2), we also compute and plot the relative point density at every point on each axis in grey. This helps us differentiate tightly bound bunches of photons from single outliers which may otherwise be disregarded.

5. Applications and Results

Our aim was to produce a tool that helps researchers to discover methods to generate view independent photon maps that are faster to render than using state-of-the-art techniques (e.g., Path traced images, Progressive Photon Relaxation or Progressive Photon Mapping) with image quality approaching that of those techniques. We have created a DSL and associated visualization toolkit to accelerate the research of photon mapping techniques. The main benefits are rapid turnaround time of ideas to code, leading to the ability to trial many approaches. Often when creating approaches, if faced with unexpected results, it is difficult to know if it is a problem of the approach, or of the code just implemented. With the DSL and visualizations we get visual feedback to determine where the problem lies. The visualizations also show how close a candidate solution is, often intimating how to develop it into a good approach.

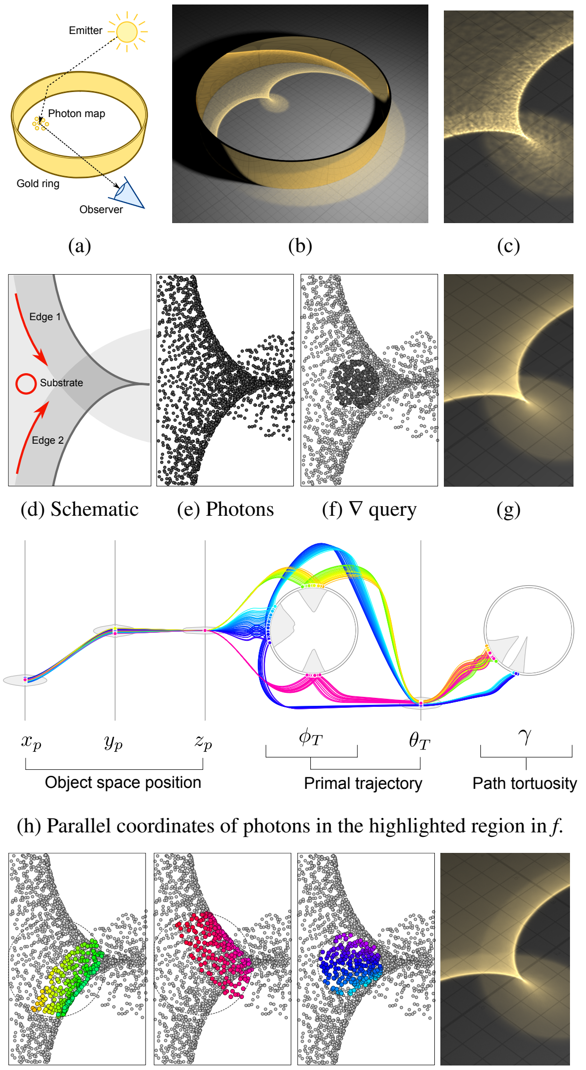
We regard research time as being the time to hypothesize and implement a new method, and in this tool we aim to reduce that considerably. Once discovered, any new methods are available as scripts in our system, and can be applied to any new scene. Sections 5.1 through 5.3 document usage of the system from the perspective of a research user. In those sections we explore a number of applications of our visualization toolkit. We demonstrate how each component works using a practical example and how they may be combined in the search for a solution to various abstruse problems surrounding parameterised photon relaxation.

Section 5.4 reports a user evaluation of the approach.

5.1. Case Study - Ring Caustic

We demonstrate the stages of our verification process using the scenario depicted in Figure 7, where a gold ring is illuminated by a single point emitter. Light striking the surface of the metal is reflected specularly onto a diffuse plane whereupon it is cached in the photon map. The relative simplicity of this setup is useful since it allows us to easily highlight features within the caustic that we wish to differentiate. Prominent among these is the internal arc that loops around the cusp of the cardioid and is visible in the close-up in 7(c).

We illustrate this structure using the schematic in 7(d). Notice that toward the centre of the figure, three components of the caustic intersect: the discontinuous edge running from top left to bottom right, its twin mirrored bottom left to top



(a) (b) (c)

(d) Schematic (e) Photons (f) ∇ query (g)

(h) Parallel coordinates of photons in the highlighted region in *f*.

(i) Parameterised gradient queries (j)

Figure 7: *(a-b) Photons are cast from a point emitter, reflected off the ring, then stored in the photon map whenever they are scattered by a diffuse surface. (c) A close-up reveals a subtle, self-intersecting band that arcs about the cusp of the cardioid. (d-f) While the eye can detect these fine structures, discrepancy in the distribution makes it difficult for the gradient estimator to differentiate between them. (g) The resulting constraints are insufficient to prevent photon relaxation smoothing away detail. (h) Deriving candidate parameterisations and visualising them with parallel coordinates reveals clusters that indicate correct structure dissociation. (i) Rebuilding the kd-tree using these new parameters and re-examination using the exploration lope confirms its effectiveness. (j) After relaxation, structure within the caustic has been correctly preserved.*

right, and the substrate layer on which both edges lie. To enable correct constraint, all three of these features should be dissociated and photons belonging to each one made unaware of intersecting layers. Not doing so leads to weak or non-existent constraints and blurring due to diffusion bias, the product of which is shown in Figure 7(g).

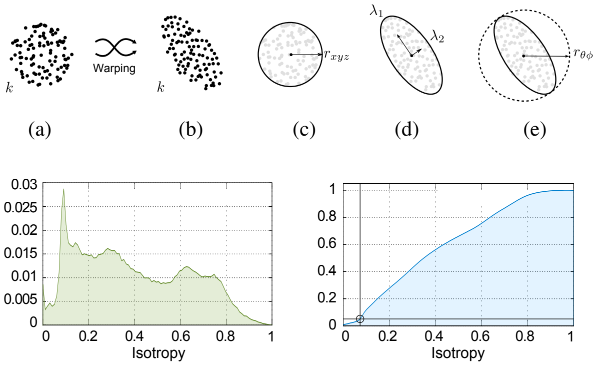
Given the point set visualisation of the structure of the distribution, we next compute a series of candidate parameterisations using the scripting language in Section 4.1. Once every photon has been successfully parameterised, we use the photon loupe to select a subset of photons in the region nearest the cursor. These data are then rendered as parallel coordinates (Figure 7(h)). Here, the coordinates from the k-NN photons highlighted in 7(f) are rendered along six perpendicular axes. The first three dimensions represent the x, y, and z spatial coordinates of the photons. Given that they are nearest-neighbors, these are grouped together into tightly-knit bunch.

The next two axes represent the primal photon trajectory from Equation 3. Immediately we observe that the main bunch branches into three separate clusters suggesting that this parameterisation is effective at separating the three filaments in the caustic marked in 7(d). The final axis encodes path tortuosity. Here we see that although this parameterisation is effective at distinguishing edges 1 and 2 from the substrate, the symmetry of the caustic distribution yields identical values of γ where the edges intersect. Without further modification, this suggests that tortuosity is not altogether effective at robust structure dissociation.

Rebuilding the kd-tree using the new parameter space allows us to verify the structure dissociation with the photon exploration loupe (Section 4.2). Figure 7(i) illustrates k-NN querying in world space, locked to a fixed range in parameter space. As the cursor moves, the restricted querying yields groups of photons whose shape matches the features we are trying to detect. The final relaxed photon map preserves the fine structure as Figure 7(j) demonstrates when compared to Figure 7(g).

5.2. Auto-Selection of Parameter β

An important consideration when implementing a new parameterisation is the degree to which the distribution is warped between world and parameter spaces. Given a disc-shaped gathering of k-NN photons queried in world space, the corresponding projection of the disc in parameter space becomes distorted (and vice versa). This warping is illustrated in Figures 8(a) and 8(b) using a small, circular set of 100-nearest neighbours in world space which is stretched into an ellipsoid in parameter space. An analogue of this phenomenon occurs in the ray differential framework [Ige99] where initially regular ray footprints become distorted as they interact with scene geometry. In the context of our application, interspatial warping necessitates enlarging the search radius in parameter space by a factor of β so as to avoid undesirable clipping and exclusion of photons (Figure 8(e)).



(a) (b) (c) (d) (e)

(f) Relative frequency

(g) Cumulative frequency

Figure 8: *(a) A circular set of k-nearest neighbouring photons in world space used to estimate the flux density gradient. (b) The same set of photons projected into parameter space. Note the warping due to the geometry transformation. (c-d) Footprint area per unit density is constant for both object and parameter space projections. (e) To guarantee a circular profile in world space, the search radius in parameter space must be large enough to account for this distortion. (f) The degree of isotropy for a given photon is equal to λ_2/λ_1. This graph shows the relative frequency histogram of isotropy for the Prism scene in Figure 9. (g) The cumulative frequency histogram allows up to estimate a value of β which minimises warp clipping yet still provides good structure differentiation.*

This enlargement is scaled by the parameter β discussed by Spencer and Jones [SJ13a] and is set to a default value of three. The optimum value of β is specific to both the scene and chosen parameterisation. Set too small and the resulting parameter space clipping leads to incorrect gradient estimates. Too large and structure dissociation becomes less effective owing to fewer photons being excluded from the search. We demonstrate how our tool can be used to analyse parameter space distortion and assist the discovery of an automatic method selecting a good value of β.

Our first step is to define a simple, scalar metric for interspatial distortion. This is accomplished by gathering the k-NN photons in parameter space and computing the principal components [Pea01] of their world space coordinates. This is done using a script and two API calls: `getKNN(i)` and `decomposePCA2D()`. The first call returns the indices of the k-nearest neighbours to photon i. The second call computes the principal components of a 2-dimensional point set.

The ratio of the square roots of the eigenvalues, $\lambda_{1,2}$, of the decomposition yields a measure of isotropy, $\imath$.

$$\imath(i) = \sqrt{\frac{\lambda_2(i)}{\lambda_1(i)}} \tag{5}$$

The histogram of $\imath$ for every photon in the Prism scene (Figure 9) is plotted in Figure 8(f). From this diagram it is immediately apparent that there is no non-zero minimum isotropy for this distribution, so merely deriving β from the smallest value of $\imath$ would not be practical. Instead, we consider a threshold value which would correctly capture a majority of photons. The cumulative distribution function for $\imath$ is plotted in Figure 8(g). Assuming we accept an error margin of 5% we obtain a value, $\imath_\beta$, above which the remaining 95% of photons fall into, is approximately $\imath_\beta = 0.075$.

Finally, we re-express β as a function of $\imath_\beta$. Assuming the area of k-NN footprint remains constant regardless of the degree of warping, we arrive at the expression:

$$\beta(i) = \sqrt{\frac{1}{\imath_\beta(i)}} \tag{6}$$

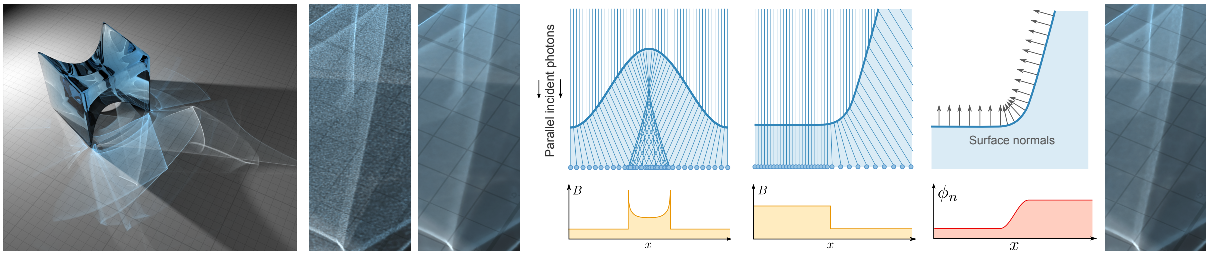
For the value of 0.075 from the Prism scene, Equation (6) yields a value of $\beta(i)$ equal to 3.65. Finding $\imath_\beta$ for an arbitrary distribution and parameterisation can be done by computing the isotropy CDF from a subset of photons, or progressively using a Monte Carlo method.

5.3. Designing a Better Parameter Space

Spencer and Jones [SJ13a] discuss a scheme based solely on the primal trajectory of each photon is used to demonstrate the effectiveness of parameterised structure differentiation. This candidate was chosen because of its theoretical consistency with models of caustic ray envelopes (illustrated in Figure 9(b)), and through experimental validation using the visualisation tool. Figure 10(a) shows the primal trajectory of each photon plotted on polar axes. Note that the silhouette of the prism is clearly visible given that this space effectively encodes a "snapshot" of the caustic-generating object from the position of the light source.

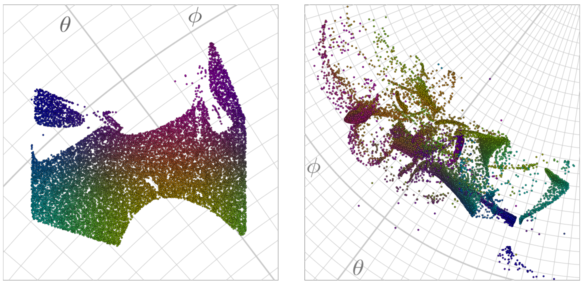
Close inspection of photon maps relaxed using primal trajectory constraints reveals that certain discontinuous transitions in flux density are not correctly dissociated by the parameterisation. A rendered example is given in Figure 9(b), where a thin stripe runs from the top to the bottom of the image. After relaxation, photons have incorrectly migrated across its leftmost edge resulting in diffusion bias and characteristic blurring (Figure 9(c)).

Figure 9(d) illustrates why these specific features remain unconstrained. Here, collimated rays pass into a curved, dielectric medium where they are refracted before being absorbed by a diffuse plane. The interface is curved in such a way so as to act as a lightguide, partitioning the photons into two discrete regions whose densities are separated by a jump discontinuity. Photons lying on this discontinuity are adjacent in parameter space, so the structure detector receives no help in resolving gradient ambiguities. Using the toolkit, we now consider whether a new parameterisation can be developed which does not suffer from these residual ambiguities.



(a) Finished render (b) Unrelaxed (c) PT (d) Envelope discontinuity (e) Bevel discontinuity (f) Surface normals (g) PPCA

Figure 9: *(a) A glass prism on a plane illuminated by a point light source. (b) A close-up of the caustic in frame a. (c) After relaxation, primal trajectory constraints (PT) have failed to completely preserve the structure of the caustic because of the bevelled edges on the prism. (d) Edges formed by ray envelopes are correctly dissociated by parameterising photons according to their origin (analogous to primal trajectory). (e) However, beveled edges create a rare discontinuous profile that is not differentiated by this parameterisation. (f) Surface normal variation near the discontinuity hints at a possible new candidate space. (g) The parameterisation based upon the principal components of the photon ray path (PPCA) is more effective.*



(a) Primal trajectory (b) Path PCA

Figure 10: *Projections of two candidate parameter spaces for the Prism scene in Figure 9. (a) The space of photon primal trajectories defined by Spencer and Jones [SJ13a]. (b) The amalgamation of photon path space vertices and normals creates a novel parameter space which is robust to unusual discontinuities.*

Ideally, the degree of variation in parameter space should reflect the rate of change in the intensity of the flux density function, B. Based on the schematic in Figures 9(d-e) we observe that photons on and around the density discontinuity show variation in their surface normals at the dielectric interface. This property is illustrated in Figure 9(f) where the relative orientation, ϕ_n, of the surface normal vector, $\vec{n}$, is plotted against its spatial position, x. This visualization leads us to consider whether the space of path normals would make a good candidate to handle the bevel discontinuity.

Since photon primal trajectory has already been shown to be effective at dissociating ray envelope discontinuities, we do not wish to entirely disregard this parameterisation. Instead we opt for an amalgamated space of the primal trajectory and the surface normals of the photon path (illustrated in Figure 3). For a given photon, the normalised primal trajectory and surface normal vectors form a cloud of points that we map into polar coordinates. From these data we calculate the distribution's moment which yields a quantitative, 2-dimensional measurement of its shape.

We found the first moment (arithmetic mean) of the distribution to be no better than primal trajectory. Its second moment, however, was more effective. Decomposition using principal components analysis yields an orthogonal basis, $[\vec{P}_1, \vec{P}_2]$, the first element of which defines the plane of maximum variance. The position of each photon in the new parameter space thus becomes $\vec{P}_1$. The photons show greater relative displacement than primal trajectory alone because more data about the photon ray path has been considered when calculating their positions. The rendered result of the new parameterisation is shown in Figure 9(g). Note how the vertical band, previously degraded by the relaxation, has been successfully detected and is now correctly constrained. The script used to generate this parameterisation is listed in Algorithm 2. In Figure 10(b), our new amalgamated parameterisation is plotted on polar axes. This demonstrates the wider displacement compared to the original primal trajectory alone. We also refer to the supplementary video for a live example of this diagnostic process using parallel coordinates.

5.4. User Experience

User Evaluation Approach. Since we are introducing a visualization system we validate the approach by conducting a small field study [Mun09]. We have already demonstrated in the prior sections the utility and success of our method in its target domain of aiding photon mapping research through the DSL and visualizations. Those previous sections provide evidence of using the system to solve three different problems and extend the research in the domain. In this section we test our approach empirically. We wish to measure how long it takes to train to use the system, with our hypothesis that a C like DSL and the accompanying visualizations will result in fast familiarization. We also want to measure how long it takes to go from hypothesis to a working solution using the system since this is the target use.

Since we aim this system at graphics researchers, we recruited three Graphics and Visualization PhD students with self reported high levels of programming ability. We measured the time to use the system and report the level of engagement with the system by the users in the study.

User Evaluation Design. We gave each participant around 15 minutes of training. This involves familiarizing the user with the DSL and visualizations using the prism caustic photon data set (Figure 9). Training includes examples of the DSL which can be loaded into the system, instruction on how to instrument the DSL to enable parameter visualization and demonstration of the visualizations within the user interface.

We then described the photon parametrization and relaxation approach, and made participants aware that their main task will be to think of their own approach for parameterising photons, implement it using the DSL and verify their approach using visualizations. They were told that unconstrained relaxation will blur edges, and shown Figures 7(a-g) to describe the problem. They were given, as an ideal target, Figure 7(j). Before moving on to the main task, we asked participants to use the system to generate images using the few pre-existing DSL examples and to study the code in the examples. The data set used for this and the main part was the ring caustic in Figure 7. We set a time limit of two hours to the entire session (including training time).

The above design allows us to test how long it takes to go from hypothesis to a working solution including training time. Since it takes days or weeks to code, debug and explore new hypotheses, anything achieved within the two hour limit will make a good case for this approach. Since participants have not seen the system prior to the user study, if they are able to generate a reasonable working solution within the two hour limit, then we know that training time and the time to implement a hypothesis is fast, even for novice users.

User Evaluation Goals. The user study goals are to determine whether the system is usable for its stated purpose of allowing the fast conversion of research hypotheses into working code, and for the user to study their approach using visualization to determine if it is parameterising the photon data set well enough to enable overlapping feature disassociation.

User Evaluation Results. Participant A decided to use photon incidence as their parameterisation. They constructed a program using the DSL to augment the photon map with photon incidence. They used the photon loupe tool, parallel coordinates and rendering to demonstrate that this parameterisation effectively separated illumination structure (Figure 11). They were able to complete their approach within the two hours, demonstrating that in their case they were able to become confident with the DSL and effectively use the visualisations to implement their hypothesis. They used their remaining time to implement the path tortuosity approach (Section 3.4.1), but not quite completing implementation.

Participant B decided to use incident and extant (from light source) photon directions. They progressed in a similar manner to participant A utilising the toolkit effectively to implement their hypothesis within the time limit.

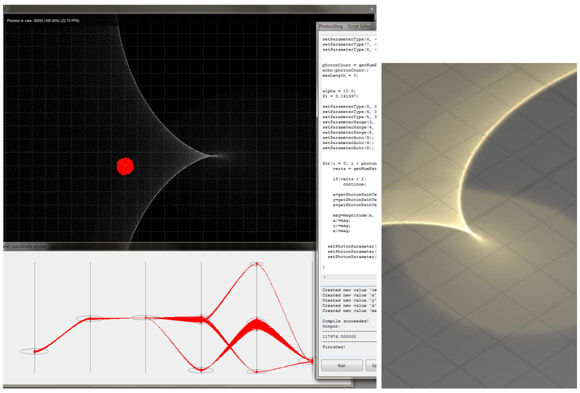
Participant C ambitiously implemented k-means clustering over photon incidence. They progressed in a similar manner to previous participants, apart from the fact that this approach was unsuccessful. This is a positive result – the tool was able to show very quickly that their approach would be a dead-end as a research direction without change.

Figure 11 demonstrates the results from Participant A. The participant wrote the code to implement photon incidence (Figure 11(a) right). They are querying the photon map with the photon loupe tool, with the parallel coordinates visualization at the bottom. In this area there are two overlapping photon sets. They were able to confirm with the visualization that their choice of parameterisation has separated the two types of photons effectively (the axes are photon position x, y, z, and vector of incidence x, y, z. A render from their approach shows they are successful in their attempt to match Figure 7(j) (also see 7(c) for the original render without relaxation). Participant B achieved similar results. Participant C's approach did not achieve the required separation, but was an interesting approach to the problem worth further consideration and development.

Participants successfully implemented their ideas. None of the participants were carrying out research in the area of photon mapping, hence they were all impressed they were able to think of an idea and transform that into a working approach within the time limit (A and B had time to spare). Participant C went beyond what we envisaged to be the capability of our system by implementing k-means clustering which demonstrates the flexibility a system employing a DSL is able to manage. All participants stated they would employ such an approach in their future research. Participant C has adopted similar visualization techniques for path tracing research, specifically for understanding algorithm design and mathematical approaches and theory prior to implementing in path tracing.

6. Conclusion

In this paper we have introduced a novel visualisation toolkit designed to allow a user to input, visualise, quantitatively assess, test and debug photon parameterisations. We have demonstrated how the use of a domain-specific language allows fast and relatively bug-free testing of candidate param-


Photons in view: 30033 (100.00%) (22.70 FPS)
allel coordinate plotter
PhotonShop - Script Editor
Run
Co

```
setParameterType(6, -
setParameterType(7, -
setParameterType(8, -


photonCount = getNumP
echo(photonCount);
maxLength = 0;


alpha = 10.0;
Pi = 3.141597;

setParameterType(3, 0
setParameterType(4, 0
setParameterType(5, 0
setParameterRange(3,
setParameterRange(4,
setParameterRange(5,
setParameterAuto(3);
setParameterAuto(4);
setParameterAuto(5);


for(i = 0; i < photon
    verts = getNumPat

    if(verts < 2)
        continue;

    x=getPhotonPathVe
    y=getPhotonPathVe
    z=getPhotonPathVe

    mag=magnitude(x,
    x/=mag;
    y/=mag;
    z/=mag;


  setPhotonParameter(
  setPhotonParameter(
  setPhotonParameter(

}
```

```
Created new value 've
Created new value 'x'
Created new value 'y'
Created new value 'z'
Created new value 'ma

Compile succeeded!
Output:
--------------------
117976.000000
--------------------
Finished!
```

Figure 11: *Screenshot from participant A (a) Photon loupe and parallel coordinates view. (b) Rendered with participant A's constraint (compare to Figures 7(c) and 7(j).*

Algorithm 2 DEFINEPPCA()

```
 // Define dimensions
X = 0; Y = 1; Z = 2; U = 3; V = 4;

declarePhotonParameter(U, -PI, PI);
declarePhotonParameter(V, -PI, PI);

for(i = 0; i < getNumPhotons(); i++)
{
 if(getNumPathVertices(i) < 2) continue;

  resetPCA2D();

  // Primal trajectory
  x0 = getPathVertex(i, 1, X) - getPathVertex(i, 0, X);
  y0 = getPathVertex(i, 1, Y) - getPathVertex(i, 0, Y);
  z0 = getPathVertex(i, 1, Z) - getPathVertex(i, 0, Z);

  mag0 = magnitude(x0, y0, z0);
  x0 /= mag0; y0 /= mag0; z0 /= mag0;

  theta = acos(z0);
  phi = atan2(x0, y0);

  pushPCA2D(theta * sin(phi), theta * cos(phi));

  // Path normals
  for(j = 1; j < getNumVertices(i); j++)
  {
    xn = getPathNormal(i, j, X);
    yn = getPathNormal(i, j, Y);
    zn = getPathNormal(i, j, Z);

    theta = acos(zn);
    phi = atan2(xn, yn) * 2;

    pushPCA2D(theta * sin(phi), theta * cos(phi));
  }

  if(decomposePCA2D() == false) continue;

  lambda = sqrt(getPCAEigenvalue(0));
  u = getPCAEigenvector(0, X) * lambda;
  v = getPCAEigenvector(0, Y) * lambda;

  setPhotonParameter(i, U, u);
  setPhotonParameter(i, V, v);
}
```

eter spaces. Furthermore, our approach of closely coupling visualisation with algorithm development leads to accelerated testing and validation resulting in new insight during visual exploration. This process is validated by case studies detailing novel work developed with the tool by incorporating surface normal data with photon primal trajectory leading to improved constraints for photon relaxation and yielding the highest quality images. As further evidence for its utility in its target area, we also report the results of a user study. We test the time to train and go from hypothesis to working solution, and demonstrate it is possible to learn how to become proficient with the visualisation and DSL toolkits within a two-hour time limit sufficient to follow such a working practice.

7. Acknowledgments

The work presented in this paper was funded by EPSRC grant number EP/I031243/1 and by HEFCW on the RIVIC project.

References

[CKR*12] Chiw C., Kindlmann G., Reppy J., Samuels L., Seltzer N.: Diderot: A parallel DSL for image analysis and visualization. *SIGPLAN Not. 47*, 6 (June 2012), 111–120. 2

[CTC10] Chen L.-H., Tsai T.-C., Chen Y.-S.: Grouped photon mapping. *The Visual Computer 26*, 3 (Mar. 2010), 217–226. 2

[CvW11] Claessen J., van Wijk J.: Flexible linked axes for multivariate data visualization. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2310–2316. 3

[DSHL10] Dammertz H., Sewtz D., Hanika J., Lensch H.: Edge-avoiding a-trous wavelet transform for fast global illumination filtering. In *Proc. High Performance Graphics 2010* (2010), pp. 67–75. 2

[EJR*13] Etiene T., Jonsson D., Ropinski T., Scheidegger C., Comba J., Nonato L., Kirby R., Ynnerman A., Silva C.: Verifying volume rendering using discretization error analysis. *Visualization and Computer Graphics, IEEE Transactions on PP*, 99 (2013), 1–1. 2

[GFE*12] Gribble C., Fisher J., Eby D., Quigley E., Ludwig G.: Ray tracing visualization toolkit. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2012), I3D '12, ACM, pp. 71–78. 2

[GJL*09] Grundy E., Jones M. W., Laramee R. S., Wilson R. P., Shepard E. L.: Visualisation of sensor data from animal movement. *Computer Graphics Forum 28*, 3 (2009), 815–822. 5

[GPL*11] Geng Z., Peng Z., Laramee R. S., Roberts J. C., Walker R.: Angular histograms: Frequency-based visualizations for large, high dimensional data. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2572–2580. 3

[HBHS05] Havran V., Bittner J., Herzog R., Seidel H.-P.: Ray maps for global illumination. In *Eurographics Symposium on Rendering* (2005), pp. 43–54. 2

[HOJ08] Hachisuka T., Ogaki S., Jensen H. W.: Progressive photon mapping. *ACM Trans. Graph. 27*, 5 (Dec. 2008), 130:1–130:8. 3

[HW09] Heinrich J., Weiskopf D.: Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1531–1538. 3

[Ige99] Igehy H.: Tracing ray differentials. In *SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 179–186. 2, 9

[Ins85] Inselberg A.: The plane with parallel coordinates. *The Visual Computer 1*, 2 (1985), 69–91. 8

[Jen96] Jensen H. W.: Global illumination using photon maps. In *Proceedings of the Eurographics workshop on Rendering techniques '96* (London, UK, 1996), Springer-Verlag, pp. 21–30. 1

[Kaj86] Kajiya J. T.: The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1986), ACM Press, pp. 143–150. 1

[LAC*11] Lehtinen J., Aila T., Chen J., Laine S., Durand F.: Temporal light field reconstruction for rendering distribution effects. *ACM Trans. Graph. 30*, 4 (July 2011), 55:1–55:12. 2

[Lar10] Laramee R.: Using visualization to debug visualization software. *Computer Graphics and Applications, IEEE 30*, 6 (2010), 67–73. 2

[MJL*13] McLoughlin T., Jones M. W., Laramee R., Malki R., Masters I., Hansen C.: Similarity measures for enhancing interactive streamline seeding. *Visualization and Computer Graphics, IEEE Transactions on 19*, 8 (Aug 2013), 1342–1353. 5

[MM08] McDonnell K. T., Mueller K.: Illustrative parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1031–1038. 3

[MSRMH09] Meyer-Spradow J., Ropinski T., Mensmann J., Hinrichs K.: Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *Computer Graphics and Applications, IEEE 29*, 6 (2009), 6–13. 2

[Mun09] Munzner T.: A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov. 2009), 921–928. 11

[NH06] Novotny M., Hauser H.: Outlier-preserving focus+context visualization in parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (2006), 893–900. 3

[NJS*11] Nowrouzezahrai D., Johnson J., Selle A., Lacewell D., Kaschalk M., Jarosz W.: A programmable system for artistic volumetric lighting. *ACM Trans. Graph. 30*, 4 (July 2011), 29:1–29:8. 3

[Pea01] Pearson K.: On lines and planes of closest fit to systems of points in space. *Philosophical Magazine 2*, 6 (1901), 559–572. 10

[Pel10] Pellacini F.: envyLight: An interface for editing natural illumination. *ACM Trans. Graph. 29*, 4 (July 2010), 34:1–34:8. 3

[QCX*07] Qu H., Chan W.-Y., Xu A., Chung K.-L., Lau K.-H., Guo P.: Visual analysis of the air pollution problem in Hong Kong. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov. 2007), 1408–1415. 3

[RKRD12] Reiner T., Kaplanyan A., Reinhard M., Dachsbacher C.: Selective inspection and interactive visualization of light transport in virtual scenes. *Comput. Graph. Forum 31*, 2 (2012), 711–718. 3

[RPLH11] Rieder C., Palmer S., Link F., Hahn H. K.: A shader framework for rapid prototyping of GPU-based volume rendering. *Computer Graphics Forum 30*, 3 (2011), 1031–1040. 2

[SD12] Sen P., Darabi S.: On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Trans. Graph. 31*, 3 (June 2012), 18:1–18:15. 2

[SFES07] Schjøth L., Frisvad J. R., Erleben K., Sporring J.: Photon differentials. In *GRAPHITE '07* (New York, NY, USA, 2007), ACM Press, pp. 179–186. 2

[SJ09] Spencer B., Jones M. W.: Into the blue: Better caustics through photon relaxation. *Eurographics 2009, Computer Graphics Forum 28*, 2 (March 2009), 319–328. 1, 3

[SJ13a] Spencer B., Jones M. W.: Photon parameterisation for robust relaxation constraints. *Eurographics 2013, Computer Graphics Forum 32*, 2 (March 2013), 83–92. 1, 4, 6, 10, 11

[SJ13b] Spencer B., Jones M. W.: Progressive photon relaxation. *ACM Trans. Graph. 32*, 1 (Feb. 2013), 7:1–7:11. 3

[SNM*13] Schmidt T.-W., Novák J., Meng J., Kaplanyan A. S., Reiner T., Nowrouzezahrai D., Dachsbacher C.: Path-space manipulation of physically-based light transport. *ACM Trans. Graph. 32*, 4 (July 2013), 129:1–129:11. 3

[TFA*11] Tam G. K. L., Fang H., Aubrey A. J., Grant P. W., Rosin P. L., Marshall D., Chen M.: Visualization of time-series data in parameter space for understanding facial dynamics. *Comput. Graph. Forum 30*, 3 (2011), 901–910. 8

[ZYQ*08] Zhou H., Yuan X., Qu H., Cui W., Chen B.: Visual clustering in parallel coordinates. *Computer Graphics Forum 27*, 3 (2008), 1047–1054. 3