Stylized Caustics: Progressive Rendering of Animated Caustics

Tobias Günther[1] Kai Rohmer[1] Christian Rössl[1] Thorsten Grosch[2] and Holger Theisel[1]

[1]University of Magdeburg, Germany
[2]Technical University of Clausthal, Germany

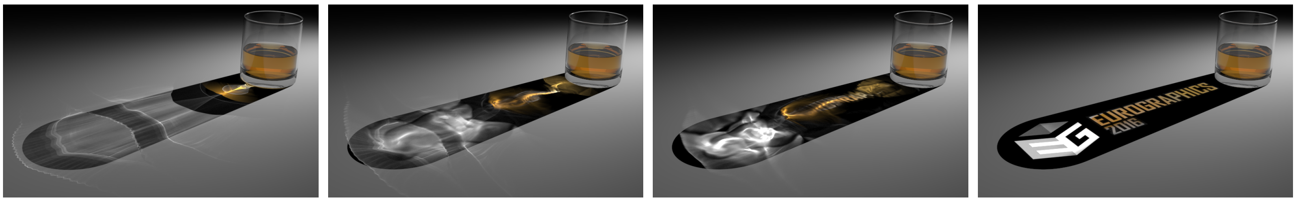


Figure 1: The original caustic (left) is blended smoothly to a user-defined target irradiance distribution (right). The depicted two intermediate frames of the animation (center) were rendered progressively and show that sharp features of the caustic are maintained.

Abstract

In recent years, much work was devoted to the design of light editing methods such as relighting and light path editing. So far, little work addressed the target-based manipulation and animation of caustics, for instance to a differently-shaped caustic, text or an image. The aim of this work is the animation of caustics by blending towards a given target irradiance distribution. This enables an artist to coherently change appearance and style of caustics, e.g., for marketing applications and visual effects. Generating a smooth animation is nontrivial, as photon density and caustic structure may change significantly. Our method is based on the efficient solution of a discrete assignment problem that incorporates constraints appropriate to make intermediate blends plausibly resemble caustics. The algorithm generates temporally coherent results that are rendered with stochastic progressive photon mapping. We demonstrate our system in a number of scenes and show blends as well as a key frame animation.

This is the authors preprint. The definitive version is available at http://diglib.eg.org/ and http://onlinelibrary.wiley.com/.

1. Introduction

Caustics are undoubtedly among the most intriguing and vibrant lighting phenomena. Produced by numerous interreflections and refractions of light within reflective and refractive material, they form strikingly beautiful and complex shapes. In consequence, a significant amount of research was devoted to the efficient simulation of caustics, see, e.g., the survey by Davidovič et al. [DKHS14]. Recent advances in light editing made it possible to manually redirect light paths [SNM*13, SMVP15] and to deform radiance signals on surfaces [RTD*10]. Yet, there is little work that automatically generates smooth and coherent animations that "blend" a caustic to an arbitrary, detailed target irradiance distribution. Ultimately, we seek to create animations of light that not only resemble natural caustics but can tell stories and subconsciously place visual cues, which are valuable tools for marketing applications and are even more vital to visual effects artists. The problem of caustic animation is challenging, since caustics usually exhibit rich structural variety and extreme density gradients. We want to see such features evolve coherently so that intermediate blends look like plausible caustics. Further, caustics are typically computed from discrete sets of photons, i.e., fairly large point sets are processed progressively.

In this paper, we propose a method that constructs a bijective map of photons from the source caustic to the target shape. This map defines for each photon a path such that its position and flux can be evaluated at different time steps of the transition between source and target to produce blends of caustics. This makes it possible to create videos of light, subtle blends between caustic structures and other fine features such as text, and caustic replacement in the spirit of *caustics by example*. The generated map satisfies two properties: First, paths should be short, i.e., unnecessary movement of photons should be penalized, and temporal evaluation should be coherent. Second, in contrast to previous work, it should preserve structures and sharp features (caustics) in intermediate blends. We model these properties by an assignment problem. Its solution leads to the minimization of a discrete energy, and we show how this can be solved efficiently. Fig. 1 demonstrates the effect.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

2. Related Work

Simulation of Light Transport

The simulation of caustics is a vital problem in photo-realistic computer graphics. First applications of caustics in graphics were pioneered by Arvo [Arv86], and Mitchell and Hanrahan [MH92]. A series of real-time techniques have been proposed based on caustic mapping [Wym05, WN09] and the rasterization of photon paths [KBW06]. Consistent light transport simulations, including caustics, strive for an exact radiance computation. Famous examples are the unbiased bi-directional path tracing [LW93] and Metropolis light transport algorithms [VG97]. Stochastic progressive photon mapping (SPPM) [HJ09] alternately traces photons and eye rays, progressively reduces the query radius of the photon density estimation, and maintains pixel statistics that eventually converge to the desired radiance, for which Knaus and Zwicker [KZ11] suggested a probabilistic reformulation. A thorough overview of GPU-based light transport was compiled by Davidovič et al. [DKHS14].

Light Editing

Recently, Schmidt et al. [SPN*15] compiled a comprehensive overview of artistic editing of appearance, lighting and material. Several hardware-accelerated relighting engines have been successfully deployed, including Lpics by Pellacini et al. [PVL*05], the linear wavelet-compressed direct-to-indirect transfer by Hašan et al. [HPB06], and the lightspeed system by Ragan-Kelley et al. [RKKS*07]. A tool for artistic control over indirect cinematic lighting was introduced by Obert et al. [OKP*08]. Pellacini et al. [PBMF07] built a global illumination interface in which an artist paints the illumination in the scene, for which matching rendering parameters are optimized. The editing of natural environment illumination was addressed with the envyLight system of Pellacini [Pel10]. Numerous techniques allow a (possibly) non-physical manipulation of light, which can be controlled directly as in the BendyLights method by Kerr et al. [KPD10] or indirectly as in Nowrouzezahrai et al. [NJS*11] for artistic volumetric lighting. Ritschel et al. [RTD*10] directly manipulate the appearance of arbitrary signals on surfaces by manual placement of handles, influence regions for deformations and sketching of B-splines. Deformations to topologically completely different distributions (e.g., caustic to text) require huge manual effort. Schmidt et al. [SNM*13] combined direct and indirect interactions, which enabled general but completely manual manipulation and retargeting of paths according to edits in the scene. They use a gizmo to interactively describe an affine transformation for the whole caustic, which makes it difficult to match given detailed target distributions. Splitting the caustic and redirecting individual path bundles toward different target patterns requires large manual effort. Subileau et al. [SMVP15] further extended the work of [SNM*13] by introducing hand-placed portals as a path-space manipulation tool. In this paper, we determine smooth movements automatically and provide more fine-grained control.

Spencer and Jones [SJ09, SJ13] moved photons to reduce noise. Gutierrez et al. [GSLM*08] heuristically recover depth from real-world photographs, establish likely caustic patterns by symmetry analysis and finally adjust luminance according to projected caustic patterns. They do not aim for smooth blendings of caustics and are not operating on a 3D scene. Klehm et al. [KISE14] optimized rendering parameters for art-directed stylization of volumetric lighting.

Caustic Fabrication

Instead of modifying caustics directly, Papas et al. [PJJ*11] and Schwartzburg et al. [STTP14] solve the inverse problem: Create a mesh that casts the desired caustic, which was considered for certain geometry templates only, e.g., planar surface on one side and directional lights. The mesh is then fabricated, i.e., caustics are static and meshes have to respect fabrication constraints. Contrary, we aim for animations and model the photon behavior during blending. Yue et al. [YIC*14] solve a correspondence problem between photons by construction of a flow. They require that correspondence maps are differentiable. Thus, blended caustics cannot overlap or fold (similar for [KP12] who state this explicitly). While this suites fabrication (single time step), this is not desirable for animated blendings.

Artistic Control and Mass Transport

Artistic control over animations of physical phenomena was successfully obtained in smoke and fluid simulations. Treuille et al. [TMPS03] pioneered the generation of target-driven smoke animations by computing guidance vector fields. Later, other approaches added local forces to the fluid motion [FL04] or applied patch-based regenerative morphing and image melding [BBRF14].

Our method establishes a map between point clouds that minimizes certain cost functions. Related to this are the *linear assignment problem* and minimization of the *Earth Mover's Distance* (EMD) in *mass transport*. Bonneel et al. [BvdPPH11] applied mass transport to blend continuous distributions, and showed results on blending discrete point sets by falling back to an assignment problem. Their method is discussed in Section 8.1. In shape correspondence, assignments are frequently formulated by an energy that preserves local structures. Mesh morphing, however, needs to consider application-specific requirements, such as changing topology and self-intersections. In contrast to methods for morphing or warping of images (which require a discretization and enforce no overlaps), the focus of caustic blending is *not* on preserving geometric properties, e.g., establishing conformal, as-rigid-as-possible, or near isometric maps. Instead, we are mainly interested in the preservation or coherent evolution of caustic-like structures such as sharp line features.

3. Motivation and Overview

The input to our method is a 3D scene, one or multiple caustics and a target irradiance distribution for each caustic to be manipulated. We apply standard progressive photon mapping, i.e., we trace and blend a new set of photons in each frame. In our method, we only consider caustic photons and displace their positions in three steps:

Projection to 2D. The core of our method processes caustic samples in a 2D domain. In a first step, we apply a perspective projection to map caustic photons from 3D onto a supporting plane and hence to 2D coordinates. Section 4 describes this step.

Blending Photon Positions. After importance sampling of the target irradiance distribution, our method establishes correspondence between source and target automatically by solving an assignment problem via minimization of a discrete energy. The energy is designed to preserve point cloud structure such that sharp line features

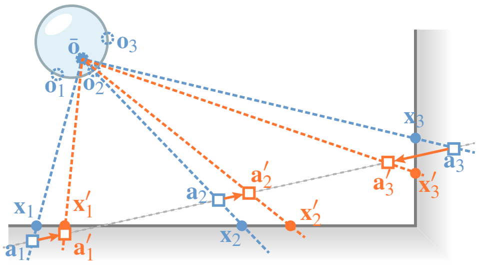


Figure 2: Projection and back projection. Photons $\mathbf{x}_i$ (●) are projected onto the supporting plane along the direction to the mean origin $\bar{\mathbf{o}}$ of the photon rays. This yields 2D positions $\mathbf{a}_i$, which are displaced to $\mathbf{a}'_i$ by blending. The photons are projected back into the scene along the orange lines, yielding 3D photon locations $\mathbf{x}'_i$ (●).

of caustics are preserved and plausibly evolve during temporal blending. Section 5.1 describes the energy and its naïve minimization, which is feasible only for small point sets. In Section 5.2 we show how larger source point sets can be efficiently assigned to target points by incorporating an interpolation and a refinement step. After assignment, we blend photon positions along cubic B-splines.

Back Projection and Rendering. Finally, we project the blended photon distribution back into the 3D scene. The procedure repeats, while the query radius of the photon density estimation progressively decreases according to [KZ11]. Section 6 gives details on the progressive rendering, and Section 7 elaborates on the implementation.

4. Projection and Back Projection

For the projection of caustic photons from $\mathbb{R}^3$ into the supporting plane, we employ a perspective projection, as illustrated in Fig. 2. The plane is progressively computed as the average of the planes spanned by the two principal components of the progressively generated photon point sets. Alternatively moving on geodesic curves is affected by local curvature distribution (e.g., bumps), which we think is undesirable. Still, it is possible to use geodesic distances in our energy, though with the inherent problems, e.g., restriction to movement in a single manifold, problems with holes, etc.

In addition to flux and position, each caustic photon stores the origin of its last traced ray. For the center of projection we use the mean of ray origins $\bar{\mathbf{o}}$. The estimate of the mean ray origin progressively improves as we average origins from all iterations. This has a favorable property: If projected photons are not altered in the plane (e.g., by a blending operation) and are "visible" from the center of projection, the projection is reversible, i.e., it results in the original caustic. Even further, the back projection of unaltered photons is independent of the choice of the supporting plane. If photons were altered or if the original 3D photons were *not* located on a planar surface, the location of the plane affects the back projected caustic. Then, we wish to minimize perspective distortion. For this, the PCA basis is a suitable choice, because if the original 3D point set was planar, the two principal components yield no distortion at all.

Given the supporting plane, the projection of $\mathbf{x}_i \in \mathbb{R}^3$ into 2D is the ray-plane intersection with the ray from the mean origin $\bar{\mathbf{o}}$ through $\mathbf{x}_i$. Representing this intersection in planar coordinates, i.e., in the basis of the supporting plane, yields $\mathbf{a}_i \in \mathbb{R}^2$. The back projection reverses this process by mapping a displaced point $\mathbf{a}'_i$, which is the result of blending $\mathbf{a}_i$ (Section 5), back into the 3D scene by casting a ray from $\bar{\mathbf{o}}$. We write the projection and back projection as $\mathbf{a}_i = P(\mathbf{x}_i)$ and $P^{-1}(\mathbf{a}'_i) = \mathbf{x}'_i$, respectively.

Target distributions are generally not aligned with the projected point set. We provide an initial alignment by mean shifting and rotating the target in the supporting plane to be upright in the scene. The target is scaled such that its three-sigma interval matches that of the source. Optionally, the user can prescribe a suitable affine transformation for adjustment. We apply a translation and scaling so that the configuration fits into the unit box $[-1,1]^2$ to ensure that regularization β (see below) is comparable for different data sets.

5. Blending Photon Sets

The projection of m photon positions into the supporting plane gives a set of points $\mathcal{A} \subset \mathbb{R}^2$ that we denote as *source*. The *target* set $\mathcal{B} \subset \mathbb{R}^2$ is obtained as a discrete importance sampling of the target irradiance with an equal number of points, i.e., $m = |\mathcal{A}| = |\mathcal{B}|$. Our goal is to achieve a blend between the two point sets: we have to construct a function that evaluates the position $\mathbf{a}'_i$ of $\mathbf{a}_i$ at time t such that $\mathbf{a}_i = \mathbf{a}'_i$ at $t = 0$ and $\mathbf{a}'_i = \mathbf{b}_j$ at $t = 1$. This holds for all $i = 1,\ldots,m$, and $j = \sigma(i)$ assigns each point in $\mathcal{A}$ a counterpart in $\mathcal{B}$. The map σ should be bijective, i.e., a permutation of indices $i = 1,\ldots,m$. This property will be useful for accelerating the computation of the assignment in Section 5.2.

5.1. Global Assignment Problem

In total, there are $m!$ possible maps σ. We are interested in maps that satisfy two properties. The first is to avoid unnecessary movement and to prefer short paths, i.e., σ should ideally minimize $\left|\left|\mathbf{a}_i - \mathbf{b}_j\right|\right|$. Using only this property leads to a standard linear assignment problem (LAP). As second requirement, we demand the preservation of point structure: We want to preserve local neighborhoods in a sense that points that are close to each other in $\mathcal{A}$ should also be close in $\mathcal{B}$. This means that intra-set distances $\left|\left|\mathbf{a}_i - \mathbf{a}_j\right|\right|$ should be preserved. We precompute the full distance matrices once to efficiently lookup individual point distances during optimization of permutation σ. Let $\mathbf{D}^{\mathcal{A}}, \mathbf{D}^{\mathcal{B}}, \mathbf{D}^{\mathcal{AB}}$ denote the distance matrices with entries

$$d_{ij}^{\mathcal{A}} = \left|\left|\mathbf{a}_i - \mathbf{a}_j\right|\right|\,, \quad d_{ij}^{\mathcal{B}} = \left|\left|\mathbf{b}_i - \mathbf{b}_j\right|\right|\,, \quad d_{ij}^{\mathcal{AB}} = \left|\left|\mathbf{a}_i - \mathbf{b}_j\right|\right|\,.$$

Then, we formulate the assignment problem as

$$\underset{\sigma}{\arg\min}\ E \ \text{ with } E = \frac{1}{m}(1-\beta)\,||\mathbf{D}^{\mathcal{A}} - \mathbf{D}^{\mathcal{B}}||_F + \beta\,\mathrm{Tr}\,\mathbf{D}^{\mathcal{AB}}$$

where $||\cdot||_F$ is the Frobenius norm, $\mathrm{Tr}\,\mathbf{D}^{\mathcal{AB}} = \sum_i \left|\left|\mathbf{a}_i - \mathbf{b}_{\sigma(i)}\right|\right|$ is the trace, and $\beta \in [0,1]$ balances preservation of relative structure versus absolute distance. (The scale $1/m$ accounts for the fact that the two terms are sums of m^2 and m values, respectively.) We are mainly interested in preserving local neighborhoods; indeed, penalization of undesired absolute movement acts mainly as a "regularizer" to prevent "flips" due to symmetries in the solution. We elaborate on the choice of β in Section 8.4.

Finding the optimal assignment σ by minimizing the objective function above yields a combinatorial optimization problem. We use a greedy algorithm to find a local minimum. Alg. 1 repeatedly applies swaps of two indices in σ. A swap is accepted if it decreases the

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

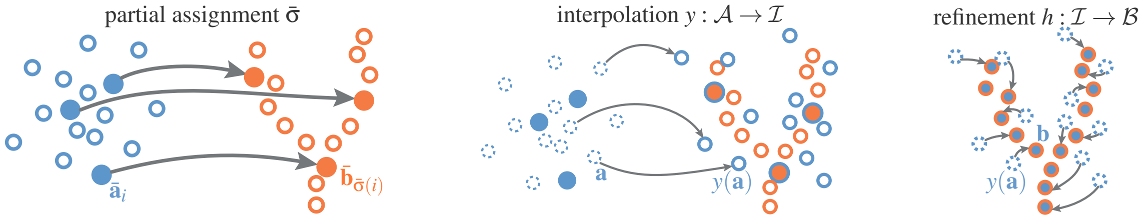

partial assignment $\bar{\sigma}$
$\bar{\mathbf{a}}_i$
$\bar{\mathbf{b}}_{\bar{\sigma}(i)}$
interpolation $y : \mathcal{A} \to \mathcal{I}$
$\mathbf{a}$
$y(\mathbf{a})$
refinement $h : \mathcal{I} \to \mathcal{B}$
$\mathbf{b}$
$y(\mathbf{a})$

Figure 3: Efficient assignment of point sets $\mathcal{A}$ and $\mathcal{B}$. First, a subset of $\mathcal{A}$ is bijectively mapped to a subset of $\mathcal{B}$ by combinatorial optimization that strives for structure preservation and short distances. Second, radial basis interpolation with constraints on the subsets transforms all points of the source caustic $\mathcal{A}$ to positions $\mathcal{I}$ close to their counterparts in the target $\mathcal{B}$. Third, the transformed points iteratively match target points by farthest closest point assignment. This refinement establishes the bijective assignment $g : \mathcal{A} \to \mathcal{B}$.

total energy E. Each swap (j,k) can be seen as swapping rows and columns j and k in matrices $\mathbf{D}^{\mathcal{A}}, \mathbf{D}^{\mathcal{B}}, \mathbf{D}^{\mathcal{AB}}$ if $j \neq k$. The algorithm is straightforward to implement using an incremental update of the current energy e: for both terms in E, first subtract contributions of rows and columns j,k and then add contributions after the swap. This way there is no need to modify the distance matrices, instead the permutation of row and column indices are modified. In this case all but the initial evaluations of E require only $O(m)$ operations.

5.2. Efficient Assignment

Optimizing assignments by Alg. 1 is costly for moderately large numbers of points m, since the algorithm requires $O(m^3)$ operations.

Subset assignment. For this reason, we apply the algorithm only on two *random subsets* $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ of $\mathcal{A}$ and $\mathcal{B}$, respectively. Both subsets have $n \ll m$ points. In all examples we use $n = 300$, which proved sufficient (see experiment in Section 8.6) and has a typical run time in the order of few seconds (<2 seconds for all examples). This yields a *partial assignment* $\bar{\sigma}$. Fig. 3 (left) illustrates the setting.

Interpolation. In the second step, we use this partial map as interpolation constraints to define a displacement function that moves all source points in $\mathcal{A}$ close to their counterparts in $\mathcal{B}$. Note that the partial map should be bijective to avoid contradicting constraints. Let $y : \mathbb{R}^2 \to \mathbb{R}^2$ denote a map of points with the interpolation property $y(\bar{\mathbf{a}}_i) = \mathbf{b}_j$ for all $i = 1,\ldots,n$ in the partial assignment $j = \bar{\sigma}(i)$. Further, we require that the map y is smooth with minimal distance distortion such that points that are close remain close after mapping. This interpolation problem can be solved easily using thin-plate splines, which minimize a linearized bending energy (see, e.g., Bookstein [Boo89]), which in turn penalizes distortion. Formally, we use radial basis functions (RBF) with kernel $\psi(r) = r^2 \ln r$ for radii $||\mathbf{x} - \bar{\mathbf{a}}_i||$ w.r.t. centers $\bar{\mathbf{a}}_i$. The interpolating function is

$$y(\mathbf{x}) = \sum_{i=1}^{n} w_i \psi_i(\mathbf{x}) + Q(\mathbf{x}) ,$$

where $\psi_i(\mathbf{x}) = \psi(||\mathbf{x} - \bar{\mathbf{a}}_i||)$ and Q is a quadratic bivariate polynomial. The weights w_i and the coefficients of Q are determined by solving a linear system of $m+6$ rows and columns. We refer to Botsch and Kobbelt [BK05] for details on setting up the system; they use RBF similarly for shape deformation.

```
Input: point sets $\mathcal{A}$, $\mathcal{B}$ with $m$ points
Output: assignment $\sigma$ given as permutation vector
$\sigma = (1,\ldots,m)$ ;        // initial assignment: identity
$e = E(\sigma)$ ;                             // minimize this
do
    for $j = 1 \ldots m$ do
        for $k = 1 \ldots m$ do
            $\sigma_{\mathrm{new}} \leftarrow \sigma \circ \mathrm{swap}(j,k)$ ;     // tentative swap
            $e_{\mathrm{new}} \leftarrow E(\sigma_{\mathrm{new}})$ ;       // energy after swap
            if $e_{new} < e$ then
                $e = e_{\mathrm{new}}$;  $\sigma \leftarrow \sigma_{\mathrm{new}}$ ;      // accept swap
            end
        end
    end
while there were swaps accepted;
```

Algorithm 1: Greedy assignment. The basic operation is a *swap* of indices in σ. The current energy e can be updated incrementally.

Refinement. Interpolation by y maps all source points $\mathbf{a}_i \in \mathcal{A}$ to positions on (for $\mathbf{a}_i \in \bar{\mathcal{A}}$ due to interpolation) or close to (due to minimal "bending" of y) positions of points in $\mathcal{B}$. This means the image $\mathcal{I} = y(\mathcal{A})$ is a point set that does not match $\mathcal{B}$ exactly but is fairly "close". Fig. 3 (center) illustrates the situation. More importantly, due to the comparatively small subset size, image $\mathcal{I}$ still contains many high-frequency details (caustics) of the source, which we can make use of later. We construct a bijective map $h : \mathcal{I} \to \mathcal{B}$ by "refining" y. Fig. 4 visualizes the error in the approximate matching without refinement. We use a farthest point sampling variant that first finds for every point in $\mathcal{I}$ the *closest point* in $\mathcal{B}$. We then iteratively select the pair with *largest distance*, remove the points from the sets, update the closest distances of the remaining points, and repeat the procedure until all points in $\mathcal{I}$ have a match. Alg. 2 describes this formally. We retrieve the point that is closest to $\mathbf{i} \in \mathcal{I}$ as closest$(\mathcal{B},\mathbf{i}) \in \mathcal{B}$, which is implemented as search in a balanced kd-tree of $\mathcal{B}$. The candidate matches and their distances are stored in a heap for fast access of the match with largest distance. Once a match is found, the selected point in $\mathcal{B}$ is removed from the kd-tree (rebalancing does not pay off), and the heap of candidate matches is updated. Fig. 3 (right) depicts the effect of refinement.

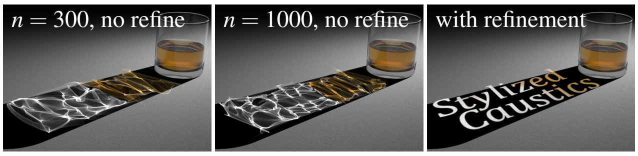


Figure 4: Impact of refinement on target positions $\mathcal{B}$. The images show blends for $t = 1$, i.e., reproduction of target. The left and center images show RBF-interpolated locations $\mathcal{I}$ *without* refinement. In the right image refinement was used to reach the target exactly.

```
Input: point sets $\mathcal{I}$, $\mathcal{B}$
Output: bijective assignment $h : \mathcal{I} \to \mathcal{B}$
while $\mathcal{I} \neq \emptyset$ do
    $\mathbf{i}' \leftarrow \arg\max_{\mathbf{i} \in \mathcal{I}} \|\mathbf{i} - \text{closest}(\mathcal{B}, \mathbf{i})\|_2$ ;          // source
    $\mathbf{b} \leftarrow \text{closest}(\mathcal{B}, \mathbf{i}')$ ;                                  // target
    $h \leftarrow h \cup (\mathbf{i}', \mathbf{b})$ ;                                           // store pair
    $\mathcal{I} \leftarrow \mathcal{I} \setminus \{\mathbf{i}'\}$; $\mathcal{B} \leftarrow \mathcal{B} \setminus \{\mathbf{b}\}$ ;                       // remove points
end
```

Algorithm 2: Farthest closest point assignment.

5.3. Summary: Blending

The partial assignment $\bar{\sigma}$ specifies constraints on the interpolation $\mathcal{I} = y(\mathcal{A})$. This interpolation function is computed *once* and is reused for all further progressively generated photon sets. Thus, in each iteration, photons are traced, projected and interpolated, followed by the farthest point assignment $\mathcal{B} = h(\mathcal{I})$ to a new set of target samples, which provides the final map $g(\mathcal{A}) = \mathcal{B}$ and likewise the global assignment $j = \sigma(i)$ such that $\mathbf{a}_i = \mathbf{b}_j$. Fig. 3 summarizes the different stages. We use the *global* bijective map

$$g : \mathcal{A} \to \mathcal{B} \quad \text{with} \quad g = y \circ h \,,$$

to find photon paths from $\mathbf{a}_i$ to $g(\mathbf{a}_i)$. Thereby, we also consider the intermediate points $y(\mathbf{a}_i)$, which are close to the target, but still contain the high-frequency structures (caustics) of the source. Let $p_{\mathbf{a}_i}(t)$ be the linear blend between source and target:

$$p_{\mathbf{a}_i}(t) \;=\; (1-t)\,\mathbf{a}_i + t\,g(\mathbf{a}_i) \quad \text{for all } i = 1,\ldots,m \,, \tag{1}$$

with time $t \in [0,1]$, and let $q_{\mathbf{a}_i}(t)$ denote the cubic B-spline that interpolates $\mathbf{a}_i, y(\mathbf{a}_i), g(\mathbf{a}_i)$ with chordal parametrization and natural boundary conditions. We define the blending operation as

$$\mathbf{a}_i' \;=\; (1-w)\,p_{\mathbf{a}_i}(t) \,+\, w\,q_{\mathbf{a}_i}(t) \quad \text{for all } i = 1,\ldots,m \tag{2}$$

and thereby introduce a control parameter $w \in [0,1]$ that balances between the linear, i.e., shortest, paths ($w = 0$) and preservation of original caustic patterns ($w = 1$). From the experiments, we recommend $w = 1/2$. Different choices are shown in Section 8.4.

6. Progressive Caustics

In this section, we explain the progressive caustic image synthesis. In each iteration, we trace photons from the light sources and regard only those as caustic photons that have either hit glass or (curved) mirror surfaces. These caustic photons have position $\mathbf{x}_i$ and flux $\phi_i^{\mathcal{A}}$. We project 3D photon positions $\mathbf{x}_i$ to 2D coordinates $\mathbf{a}_i = P(\mathbf{x}_i)$ in the support plane (see Section 4). Then, we bijectively map photons to their sample of the target distribution and apply linear interpolation (2) between the linear and the B-spline path. This gives

$$\mathbf{x}_i' = P^{-1}\left((1-w)\,p_{P(\mathbf{x}_i)}(t) + w\,q_{P(\mathbf{x}_i)}(t)\right) \tag{3}$$

$$\phi_i = (1-t)\,\phi_i^{\mathcal{A}} + t\,\phi_{\sigma(i)}^{\mathcal{B}} \tag{4}$$

for the displacement of a photon position $\mathbf{x}_i$ to its final back projected destination $\mathbf{x}_i'$. The flux ϕ_i of the photon is a blend between flux in source $\mathcal{A}$ and target distribution $\mathcal{B}$.

In each iteration, we compute the approximate radiance $\hat{L}$ at location $\mathbf{x}$ in direction ω by the radiance estimate [Jen01]:

$$\hat{L}(\mathbf{x},\omega) = \sum_i \phi_i\,\kappa(\|\mathbf{x}_i' - \mathbf{x}\|)\,f_r(\mathbf{x},\omega,\omega_i) \,. \tag{5}$$

Thereby, $f_r(\mathbf{x},\omega,\omega_i)$ is the BRDF, characterizing the amount of light reflected from a photon with incoming direction ω_i at position $\mathbf{x}$ toward the viewer in direction ω. Further, $\kappa(s)$ is a density falloff function with unit integral that weights the contribution of a photon in a certain query radius. For a simple unit disk with radius r, we set $\kappa(s) = \frac{1}{\pi r^2}$ if $s \leq r$ and zero otherwise.

Using stochastic progressive photon mapping [HJ09], we let the estimates converge to the consistent solution:

$$L(\mathbf{x},\omega) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} \hat{L}(\mathbf{x},\omega) \tag{6}$$

Thereby, the query radius r of the kernel $\kappa(s)$ decreases over time. In the present work, we used a global radius and shrinked it by the factor $\sqrt{(k+\alpha)/(k+1)}$ as suggested by Knaus and Zwicker [KZ11] for the k-th iteration with reduction rate $\alpha = 2/3$.

Note that $t = 0$ in Eqs. (3) and (4) yields the original caustic, i.e., $\mathbf{a} = P^{-1}(P(\mathbf{a}))$, if all caustic photons are "visible" from the center of projection. Setting $t = 1$ yields a direct sampling of the target distribution, i.e., the target is reached exactly.

7. Details and Implementation

Caustic Photon Generation and Rendering

We implemented the caustic photon tracing on the GPU using Nvidia's raytracing API OptiX [PBD*10]. The irradiance estimates are implemented by splatting with a progressively decreasing kernel radius [KZ11]. Direct illumination is obtained by ray casting.

We distinguish between light paths based on their recent scene interaction events in the spirit Schmidt et al. [SNM*13]. We associate each deposited caustic photon with a source ID that uniquely encodes which light source the photon originated from and which object it hit last. This ID segments photons into subsets, each representing an individual caustic. Fig. 5 for example, shows a glass sphere and two lights. The lights create two caustics that can be treated separately. If source-based segmentation does not prove sufficient, point sets could optionally be further merged or subdivided.

Selection of Target Distribution

The target distribution essentially specifies the resulting irradiance, and it is thus typically given as a gray-scale image or by luminance. It may contain, e.g., text, other caustics, drawings or photographs. For the sampling of positions $\mathcal{B}$ in static target images, we use the inverse CDF method [PH10]. It samples a certain probability distribution by inverting its cumulative distribution function (CDF), which maps a uniform random variable to the target distribution.

We also experimented with animated target irradiance distributions. Here, care must be taken to ensure that in each frame of the

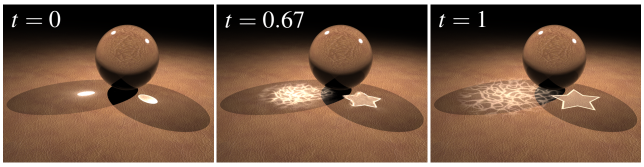


Figure 5: A glass sphere is illuminated by two light sources, which cast two separate caustics. As we distinguish between photon origins, they can be processed individually.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

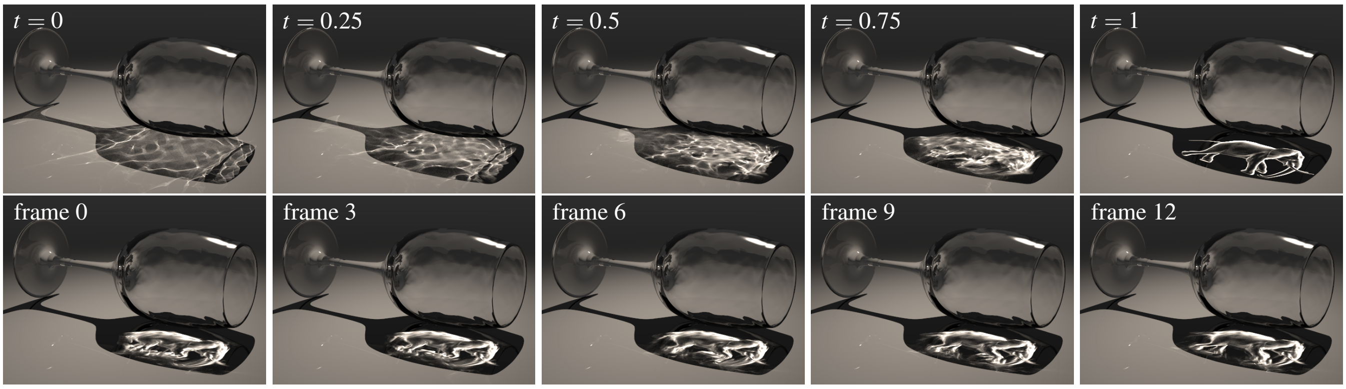


Figure 6: Top row: A caustic is blended into an elephant. Bottom row: The target irradiance is chosen from a key frame animation. Each frame is shown at $t = 0.9$, i.e., the caustic has not yet reached the target shape. See the accompanying **video** for further examples.

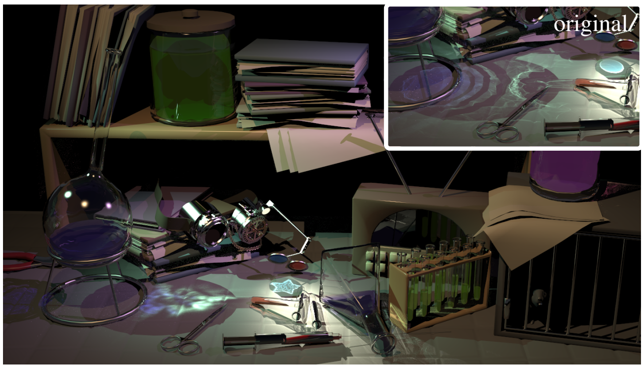


Figure 7: Large scene with two caustics replaced on the ground.

video, samples $\mathcal{B}$ are generated frame-coherently, e.g., in the spirit of [WMWL11]. That is, caustic photons $\mathcal{A}$ (which were traced deterministically), should be mapped to a semantically similar location in $\mathcal{B}$, so that the caustic will blend toward the same object part throughout the video. To obtain this coherence requires the identification of corresponding feature points. The animation in Fig. 6 shows a rendered video of a key frame animated elephant with already established vertex correspondence, i.e., the elephant can be animated by blending its vertices. That is, every vertex keeps its semantic over the course of the animation. In the first frame of the animation, we selected 400 vertices by geodesic farthest point sampling and kept those as feature points throughout the animation. Frame coherence is achieved fairly well already, though the crossing of the legs during the walk cycle leads to conflicting conditions in the interpolation.

8. Results and Discussion

We tested our method in a number of scenes using different target distributions. The blending to text and logos was shown in the WHISKEY scene in Fig. 1 and Fig. 4. In Fig. 14, we rotate the target distribution prior to the assignment such that orange photons move to the top of the text. Further, we used drawings as target, for instance, for the caustics of a SPHERE in Fig. 5 and in the more complex LAB scene in Fig. 7. We also demonstrated an animation of the target distribution to animate the walk cycle of an elephant in a GLASS caustic, see Fig. 6. (See the accompanying video.)

8.1. Comparison with Other Assignments

We compare our assignment algorithm (Section 5.1) to two alternative strategies: solving the *linear assignment problem* (LAP) and minimizing the *Earth Mover's Distance* (EMD). The LAP establishes a bijective map that minimizes the traveled distance. It is the special case of our algorithm for $\beta = 1$, i.e., *without* enforcing preservation of structure. The LAP can be solved by the Hungarian method, also known as the Kuhn-Munkres algorithm. Mass transport minimizes the EMD, which allows points to split up their mass (flux) while traveling. Bonneel et al. [BvdPPH11] propose a general, multi-scale method for blending continuous distributions by representing them as a sum of radial basis functions (with Gaussian kernel), and moving them via mass transport. For the refinement in the efficient assignment (Section 5.2) we require a bijective map to formulate interpolation conditions for displacement by thin-plate splines (RBF with harmonic kernel). Generally, mass transport does not give a bijective map, as mass (flux) is split among samples.

Both LAP and EMD minimize a distance measure between source and target. In contrast to our method, they do not explicitly enforce structure preservation. We demonstrate this effect for blending (with $w = 0$) in Fig. 8, which shows the transition of a C-shaped caustic to the letter S, and vice versa. While EMD allows particles to split and merge (not bijective), both LAP and EMD minimize travelled distance, and thus they deliver comparable results: Parts of the letters break out. Our algorithm penalizes this behavior for $\beta > 0$.

Moreover, matching large photon sets by LAP or EMD in each light transport iteration is not feasible in practice, since each iteration may take minutes to compute. Fast approximations to EMD by thresholded ground distances [PW09, LML*14] yield arbitrary assignments for distant particles [BvdPPH11] and thus appear blurry in a progressive renderer. Another acceleration of EMD is regularized transport [Cut13], which, for instance, increases the number of non-zero couplings. Still, for progressively generated photon sets, the resulting blur depends on the photon density in the scene. Since minimizing EMD does not generally yield a bijective map, mass transport cannot be used on a subset with our subsequent RBF interpolation for efficient assignment, which would generate another path control point. Thus, EMD only uses linear paths. For $w > 0$, our method blends along nonlinear paths, which makes use of high-frequency details contained in $\mathcal{I}$ (see Section 5.2). This significantly improves the intermediate caustic patterns, as shown later in Fig. 11.

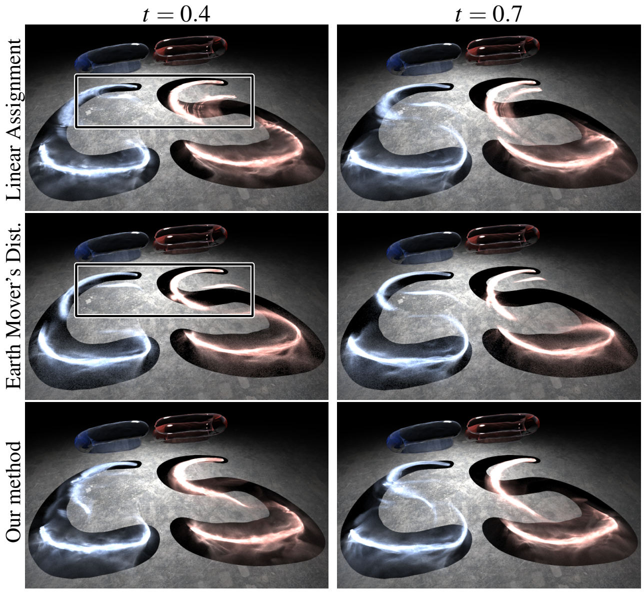


Figure 8: Comparison between linear assignment (first row), Earth Mover's Distance (second row) and our method (third row) for $w = 0$. In this scene, we blend a C-shaped caustic to an S, and vice versa. Our method explicitly enforces structure preservation, thus parts of the letters are not breaking out, see highlighted areas.

8.2. Comparison with Other Refinement Strategies

In this section, we compare our *refinement* algorithm (Section 5.2) with three alternative strategies. The goal is to map the caustic photons after assignment and interpolation $\mathcal{I} = \{\mathbf{i}_i\}$ to the importance-sampled target positions $\mathcal{B} = \{\mathbf{b}_i\}$ to establish a global assignment.

Random Permutation Assignment. The most trivial matching of $\mathcal{I}$ with $\mathcal{B}$ is a random assignment. Since both sets are of equal cardinality and both are obtained by random sampling, a random permutation is obtained by $h(\mathbf{i}_i) = \mathbf{b}_i$. Unfortunately, the random assignment leads to very blurry results that quickly lose caustic-like appearance, see the first column of Fig. 9.

Inverse CDF-based Assignment. A different approach is an assignment based on the inverse CDF method. The user already provides a target distribution function. The idea is to additionally generate a distribution function of the transformed photons $\mathcal{I}$. Such a distribution function is essentially obtained by progressive radiance estimates on a discretized grid in the 2D plane. Given the two distributions, CDFs are computed for both. Then using the inverse CDF method, samples are not only generated for the target distribution to obtain $\mathcal{B} = \{\mathbf{b}_i\}$, but also for the estimated source distribution *using the same uniform random variables* ξ to obtain a set $\mathcal{I}' = \{\mathbf{i}'_i\}$, which has the same distribution as $\mathcal{I}$. Then, the bijection for blending becomes $h(\mathbf{i}'_i) = \mathbf{b}_i$. A disadvantage of this method is that it requires to (progressively) estimate a *discretized* source distribution function of the transformed point set $\mathcal{I}$ to obtain its CDF. (We used a 2k × 2k grid.) Furthermore, it tends to produce visual artifacts by tearing structures apart, and the inherent grid structure of the CDF remains visible during blending, as noticeable in Fig. 9.

Bidirectional Closest Point Assignment. In contrast to the other approaches, the third approach does not strive for a bijective map. First, every point $\mathbf{i} \in \mathcal{I}$ searches for its closest point in $\mathcal{B}$, i.e., $h \leftarrow h \cup (\mathbf{i}, \text{closest}(\mathcal{B}, \mathbf{i}))$. In $\mathcal{B}$ a number of points remain that have not been assigned yet: $\mathbf{b}' \in \mathcal{B} \setminus \{\text{closest}(\mathcal{B}, \mathbf{i})\}$. Each of these points searches for its closest point in $\mathcal{I}$, i.e., $h \leftarrow h \cup (\text{closest}(\mathcal{I}, \mathbf{b}'), \mathbf{b}')$. The result forms a graph with possibly multiple outgoing edges from $\mathcal{I}$ and possibly multiple incoming edges in $\mathcal{B}$. To preserve energy, the flux of transformed photons $\mathcal{I}$ is divided by the number of outgoing edges. The incoming flux in $\mathcal{B}$ is normalized to match the target flux. Since this map is not bijective, it cannot preserve the color of photons: consider, for instance, five orange photons mapped to the same target position; then only one orange photon remains visible in the resulting image. Moreover, this algorithm also tends to tear structures apart, which is a quite noticeable artifact, see Fig. 9.

In comparison to the methods above, the farthest closest point method described in Section 5.2 behaves more naturally and does not tear apart visible sharp edges, see the accompanying video.

8.3. Comparison with Mesh Morphing

In the video, we compare our method with "real" caustics obtained by morphing two spheres with different Perlin noise displacement maps. Our method produces coherent and smooth blends, though near the end of the transition caustics lose details and fade into each other, since a small subset cannot capture all fine structures. Instead, mesh morphing results in many small caustics moving uncontrollably, though with more apparent details. While in some cases the properties of our approach are desirable, "real" caustics may look more natural. Thus, if the user hopes for very realistic motion/caustics our approach may not be most appropriate.

8.4. Choice of Parameters

Our algorithm (Section 5) uses a parameter $\beta \in [0, 1]$, which balances influence of relative structure versus absolute distance. Large values of β generally do not make sense because a primarily distance-based assignment is often not meaningful and disregards structure. If, however, β is too small or even zero, then one may get undesired assignments. This happens if there exists a multitude of solutions with minimal or similarly low energy, a situation which becomes likely if there exist symmetries (in the solution) or the subset size is too small. In this case the solution may "flip", and higher β penalizes such undesired flips. Fig. 10 gives an example. Values of $4 \cdot 10^{-4} < \beta < 1 \cdot 10^{-2}$ work well and give comparable results (see additional material). If not mentioned otherwise, we set $\beta = 4 \cdot 10^{-4}$; and there is usually not much need to change this value.

Eq. (2) describes the path from source to target as a blend between a linear path and a cubic B-spline. Different choices of its weight w are shown in Fig. 11 for $\beta = 1 \cdot 10^{-3}$. In our experience, linear paths ($w = 0$) appear slightly blurred. The interpolating B-spline ($w = 1$) through the RBF-interpolated point $y(\mathbf{a}_i)$ on the other hand, retains the sharp features of the source caustic longer. We recommend the half-way blend $w = 0.5$ as default value to obtain a compromise between short paths and preservation of high-frequency features.

In contrast to progressive photon mapping, the number of photons that are traced in an iteration *does* influence the converged solution of our method. This is because, the more caustic photons are used

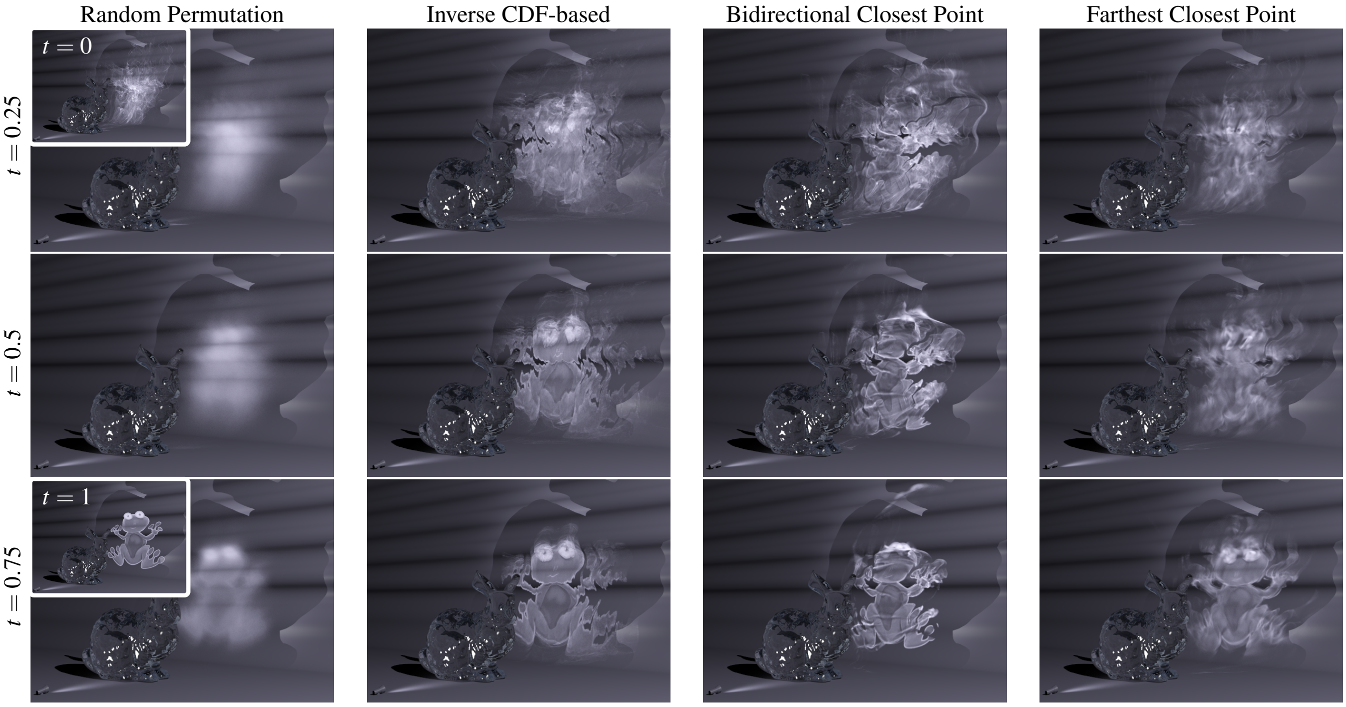


Figure 9: Comparison with other refinement strategies for $w = 0$. Rows depict different techniques at the same time with first and last frame as insets in the first column. Here, photons are cast onto a non-planar surface. The back projection does not induce visible artifacts from perspective distortion. From left to right: The random permutation yields a blurry result. The inverse CDF-based method shows locally high frequency distortion (e.g., arms). Bidirectional closest point assignment tears the frog shape. Our farthest closest point method blends smoothly.

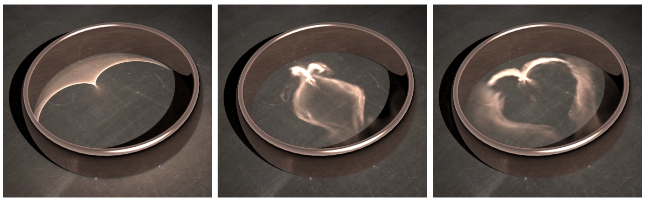


Figure 10: Parameter β trades preservation of point structure for absolute distance, which is important for symmetric data. Left: cardioid (heart) caustic in a copper ring, which we blend into an actual heart shape, for $t = 0.8$. Center: $\beta = 1 \cdot 10^{-4}$, which could not prevent a flip. Right: flip is prevented, for which we set $\beta = 2 \cdot 10^{-4}$.

the less they need to move to reach their target position during the refinement step y. As each iteration is operating on a different photon set, this movement will act as a low-pass filter, with a kernel size in the order of the average distance to the target position.

8.5. Remaining Error

Obtaining a ground truth matching between arbitrary point sets has factorial complexity, which cannot be computed for our problem sizes ($n = 300$ gives $3 \cdot 10^{614}$ candidates). We can, however, give evidence for the quality of our minimizer by constructing a benchmark with known ground truth. For this, we draw samples from a caustic image (source) and randomly permute these samples (target). A perfect assignment would undo this permutation (error $= 0$). If a random assignment is assumed as lower error bound, our method reaches $> 95\%$ of the ground truth (for $n = 300$, $\beta = 4 \cdot 10^{-3}$). This percentage further increases for higher n, as shown in Fig. 12.

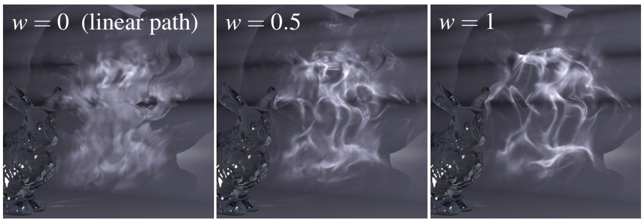


Figure 11: Different choices of w for the blending path at $t = 0.5$.

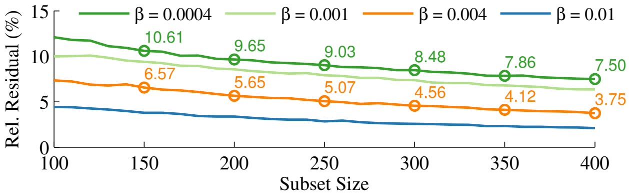


Figure 12: Plot of the relative residual between ground truth and random assignment. For higher β the energy is better minimized.

8.6. Performance

Our test system is equipped with an Intel Core i7-2600K CPU with 3.4 GHz and 24 GB RAM and an Nvidia Quadro K5000 GPU with 4 GB VRAM. The runtime of the individual steps of an iteration is listed in Table 1. The tracing of caustic photons and their eventual back projection are easily parallelizable and implemented on the GPU. Compared to the other steps, the computation of the subset assignment (Alg. 1) for the chosen subset size of $n = 300$ typically takes the longest time (< 1.5 seconds). However, this is only a preprocessing step that is executed once. Among the progressive

	Whiskey	Sphere	Bunny
	Fig. 1	Fig. 5	Fig. 9
Photon Tracing	10.69	9.67	10.10
Projection P	3.38	5.80	3.25
Subset Assignment *	1,203.51	1,228.54	1,285.23
Fit $y^{\star}$	9.37	7.26	8.65
Sample Target Set $\mathcal{B}$	0.65	1.63	2.03
Evaluate y	29.87	101.51	124.97
Target Refinement	24.48	274.16	494.38
Back Projection P^{-1}	4.12	4.27	4.38
Caustic Photons/Iter.	1,626	5,570	7,060

Table 1: Timings of algorithm steps in milliseconds for $n = 300$. Preprocessing steps marked by * are executed only once.

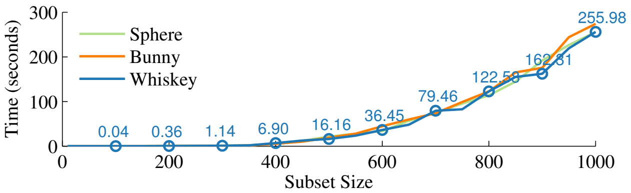


Figure 13: Runtime of subset assignment for varying subset sizes. For up to 300–400 points the runtime stays in interactive ranges.

steps especially the refinement to the target distribution is the bottleneck. Depending on the number of caustic photons in the scene, near-interactive updates can be achieved for the iterations.

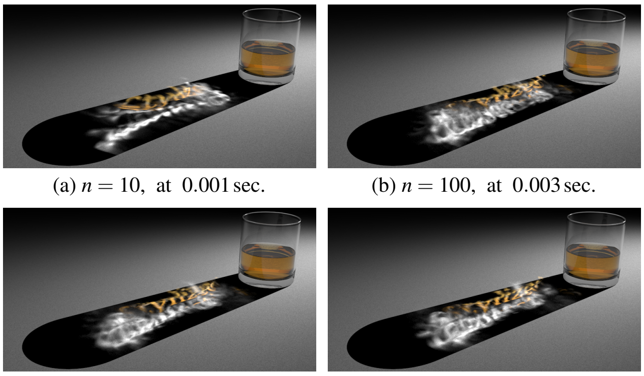
The runtime of the subset assignment is $O(n^3)$ for subset size n. We measured the duration of the assignment computation for different n in three different test scenes. As expected, the runtime scales similarly in all scenes. During interactive manipulation of scene geometry, lights or target distributions, the subset assignment has to be recomputed. Fig. 13 shows that the runtime starts to degrade at a subset size > 300–400 photons, thus we set $n = 300$, if not mentioned otherwise. As shown in Fig. 14, the quality does not improve significantly if the subset size is further increased.

8.7. Limitations

During blending between caustics and their targets, the photon density might vary over time, as particle trajectories might cross. This becomes apparent, when blending a smooth distribution into another, as here, the result is not necessarily likewise smooth, see Fig. 15 for an example. This is not a big problem, since in those cases simpler blending methods would suffice. For us, the appearance of caustic-like structures in intermediate frames is acceptable.

Currently, our method does not support dynamic source caustics, e.g., by moving caustic casting objects or lights. Depending on the magnitude of the change in the caustic, the RBF interpolation y might still be suitable, e.g., for a waving water surface. If the caustic topology changes significantly, correspondence between source distributions must be prescribed or somehow enforced to obtain temporally coherent RBF interpolations.

The projection onto the supporting plane might not be reversible on complex shapes due to occlusions. Our main applications are (near)-planar receivers, as they showcase resulting caustics best.



(a) $n = 10$, at 0.001 sec.

(b) $n = 100$, at 0.003 sec.

(c) $n = 300$, at 1.105 sec.

(d) $n = 1000$, at 224.3 sec.

Figure 14: Results for different subset size n for the Whiskey scene at $t = 0.65$. The time in the captions refers to the computation time of the subset assignment (Alg. 1). The results are quite similar even for very small n, while the matching is orders of magnitude faster.

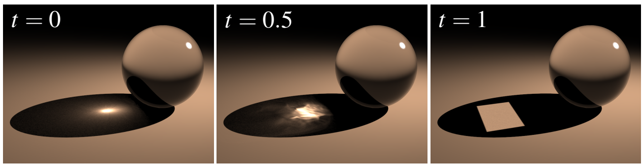


Figure 15: Limitation: blending between two smooth distributions does not necessarily result in a smooth intermediate distribution.

9. Conclusions and Future Work

We presented a method for the manipulation of caustics that blends a given caustic smoothly to a user-defined target irradiance distribution, e.g., an image, text or a video. To the best of our knowledge, this kind of manipulation and control of caustics is novel. Technically, our method requires only a target image and provides a steering parameter w that allows to retain structures of the source caustic. The core algorithm for caustic blending solves an assignment problem by minimizing a discrete energy. This energy is carefully designed such that blends evolve to structures that resemble caustics. Its minimization is not straightforward, and we show how this can be achieved efficiently using an additional interpolation and refinement step.

Future work includes processing animated scene geometry, e.g., a moving water surface, improved frame coherence if target distributions are time-dependent, and accounting for additional quantities such as photon flux in the optimization. The method could also be applied to temporal upsampling of caustics. We demonstrated our method with SPPM. Other light transport techniques such as bidirectional path tracing or vertex connection and merging would work similarly: When tracing photon sub-paths, caustic photons can be deformed by our method, and afterwards connected with eye sub-path vertices. Also, deformed photons can continue to bounce in the scene, adding full global illumination effects.

Acknowledgements We thank Robert W. Sumner & Jovan Popović for the elephant and Dan Konieczka & Giorgio Luciano for the lab scene. Supported by DFG grant GR 3833/2-1 and DA 1319/2-1.

References

[Arv86] ARVO J.: Backward ray tracing. In *SIGGRAPH '86 Course Notes – Developments in Ray Tracing* (1986), ACM, pp. 259–263. 2

[BBRF14] BROWNING M., BARNES C., RITTER S., FINKELSTEIN A.: Stylized keyframe animation of fluid simulations. In *Proc. Workshop on Non-Photorealistic Animation and Rendering* (2014), ACM, pp. 63–70. 2

[BK05] BOTSCH M., KOBBELT L.: Real-time shape editing using radial basis functions. *Computer Graphics Forum 24*, 3 (2005), 611–621. 4

[Boo89] BOOKSTEIN F. L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell. 11*, 6 (1989), 567–585. 4

[BvdPPH11] BONNEEL N., VAN DE PANNE M., PARIS S., HEIDRICH W.: Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 30*, 6 (2011), 158:1–158:12. 2, 6

[Cut13] CUTURI M.: Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 2292–2300. 6

[DKHS14] DAVIDOVIČ T., KŘIVÁNEK J., HAŠAN M., SLUSALLEK P.: Progressive light transport simulation on the GPU: Survey and improvements. *ACM Trans. Graph. 33*, 3 (2014), 29:1–29:19. 1, 2

[FL04] FATTAL R., LISCHINSKI D.: Target-driven smoke animation. *ACM Trans. Graph. (Proc. SIGGRAPH) 23*, 3 (2004), 441–448. 2

[GSLM*08] GUTIERREZ D., SERON F. J., LOPEZ-MORENO J., SANCHEZ M. P., FANDOS J., REINHARD E.: Depicting procedural caustics in single images. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 27*, 5 (2008), 120:1–120:9. 2

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 28*, 5 (2009), 141:1–141:8. 2, 5

[HPB06] HAŠAN M., PELLACINI F., BALA K.: Direct-to-indirect transfer for cinematic relighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 1089–1097. 2

[Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001. 5

[KBW06] KRÜGER J., BÜRGER K., WESTERMANN R.: Interactive screen-space accurate photon tracing on GPUs. In *Eurographics Symp. on Rendering* (2006), pp. 319–329. 2

[KISE14] KLEHM O., IHRKE I., SEIDEL H.-P., EISEMANN E.: Property and lighting manipulations for static volume stylization using a painting metaphor. *IEEE TVCG 20*, 7 (2014), 983–995. 2

[KP12] KISER T., PAULY M.: *Caustic Art*. Tech. rep., EPFL, 2012. 2

[KPD10] KERR W. B., PELLACINI F., DENNING J. D.: Bendylights: Artistic control of direct illumination by curving light rays. In *Proc. Eurographics Symposium on Rendering* (2010), pp. 1451–1459. 2

[KZ11] KNAUS C., ZWICKER M.: Progressive photon mapping: A probabilistic approach. *ACM Trans. Graph. 30*, 3 (2011), no. 25. 2, 3, 5

[LML*14] LI L., MA M., LEI P., WANG X., CHEN X.: A linear approximate algorithm for Earth Mover's Distance with thresholded ground distance. *Mathematical Problems in Engineering* (2014), 406358:1–406358:9. 6

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. *Proc. Compugraphics* (1993), 145–153. 2

[MH92] MITCHELL D., HANRAHAN P.: Illumination from curved reflectors. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 283–291. 2

[NJS*11] NOWROUZEZAHRAI D., JOHNSON J., SELLE A., LACEWELL D., KASCHALK M., JAROSZ W.: A programmable system for artistic volumetric lighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4 (2011), 29:1–29:8. 2

[OKP*08] OBERT J., KŘIVÁNEK J., PELLACINI F., SÝKORA D., PATTANAIK S. N.: iCheat: A representation for artistic control of indirect cinematic lighting. *Comp. Graph. Forum 27*, 4 (2008), 1217–1223. 2

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: a general purpose ray tracing engine. *ACM Trans. Graph. (Proc. SIGGRAPH) 29* (2010), 66:1–66:13. 5

[PBMF07] PELLACINI F., BATTAGLIA F., MORLEY K., FINKELSTEIN A.: Lighting with paint. *ACM Trans. Graph. 26*, 2 (2007), no. 9. 2

[Pel10] PELLACINI F.: envylight: An interface for editing natural illumination. *ACM Trans. Graph. (SIGGRAPH) 29*, 4 (2010), no. 34. 2

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 5

[PJJ*11] PAPAS M., JAROSZ W., JAKOB W., RUSINKIEWICZ S., MATUSIK W., WEYRICH T.: Goal-based caustics. *Computer Graphics Forum (Proc. Eurographics) 30*, 2 (2011), 503–511. 2

[PVL*05] PELLACINI F., VIDIMČE K., LEFOHN A., MOHR A., LEONE M., WARREN J.: Lpics: A hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005), 464–470. 2

[PW09] PELE O., WERMAN M.: Fast and robust Earth Mover's Distances. In *Proc. IEEE Computer Vision* (2009), pp. 460–467. 6

[RKKS*07] RAGAN-KELLEY J., KILPATRICK C., SMITH B. W., EPPS D., GREEN P., HERY C., DURAND F.: The lightspeed automatic interactive lighting preview system. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3 (2007), 25:1–25:11. 2

[RTD*10] RITSCHEL T., THORMÄHLEN T., DACHSBACHER C., KAUTZ J., SEIDEL H.-P.: Interactive on-surface signal deformation. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 4 (2010), 36:1–36:8. 1, 2

[SJ09] SPENCER B., JONES M. W.: Into the blue: Better caustics through photon relaxation. *CGF (Proc. Eurographics) 28*, 2 (2009), 319–328. 2

[SJ13] SPENCER B., JONES M. W.: Progressive photon relaxation. *ACM Trans. Graph. 32*, 1 (2013), 7:1–7:11. 2

[SMVP15] SUBILEAU T., MELLADO N., VANDERHAEGHE D., PAULIN M.: Light transport editing with ray portals. In *Proc. Computer Graphics International* (2015), p. to appear. 1, 2

[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. (Proc. SIGGRAPH) 32*, 4 (2013), 129:1–129:11. 1, 2, 5

[SPN*15] SCHMIDT T.-W., PELLACINI F., NOWROUZEZAHRAI D., JAROSZ W., DACHSBACHER C.: State of the art in artistic editing of appearance, lighting and material. *Computer Graphics Forum* (2015), to appear. 2

[STTP14] SCHWARTZBURG Y., TESTUZ R., TAGLIASACCHI A., PAULY M.: High-contrast computational caustic design. *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4 (2014), 74:1–74:11. 2

[TMPS03] TREUILLE A., MCNAMARA A., POPOVIĆ Z., STAM J.: Keyframe control of smoke simulations. *ACM Trans. Graph. 22*, 3 (2003), 716–723. 2

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), SIGGRAPH '97, pp. 65–76. 2

[WMWL11] WAN L., MAK S.-K., WONG T.-T., LEUNG C.-S.: Spatiotemporal sampling of dynamic environment sequences. *IEEE TVCG 17*, 10 (2011), 1499–1509. 6

[WN09] WYMAN C., NICHOLS G.: Adaptive caustic maps using deferred shading. *CGF (Proc. Eurographics) 28*, 2 (2009), 309–318. 2

[Wym05] WYMAN C.: An approximate image-space approach for interactive refraction. *ACM Trans. Graph. (Proc. SIGGRAPH) 24*, 3 (2005), 1050–1053. 2

[YIC*14] YUE Y., IWASAKI K., CHEN B.-Y., DOBASHI Y., NISHITA T.: Poisson-based continuous surface generation for goal-based caustics. *ACM Trans. Graph. 33*, 3 (2014), 31:1–31:7. 2

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.