Controllable Caustic Animation Using Vector Fields

Irene Baeza[1,2], Markus Gross [1,2] and Tobias Günther[1]

[1]Department of Computer Science, ETH Zürich, Switzerland [2]Disney Research Studios, Zürich, Switzerland

Abstract

In movie production, lighting is commonly used to redirect attention or to set the mood in a scene. The detailed editing of complex lighting phenomena, however, is as tedious as it is important, especially with dynamic lights or when light is a relevant story element. In this paper, we propose a new method to create caustic animations, which are controllable through constraints drawn by the user. Our method blends caustics into a specified target image by treating photons as particles that move in a divergence-free fluid, an irrotational vector field or a linear combination of the two. Once described as a flow, additional user constraints are easily added, e.g., to direct the flow, create boundaries or add synthetic turbulence, which offers new ways to redirect and control light. The corresponding vector field is computed by fitting a stream function and a scalar potential per time step, for which constraints are described in a quadratic energy that we minimize as a linear least squares problem. Finally, photons are placed at their new positions back into the scene and are rendered with progressive photon mapping.

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

1. Introduction

Artistically controlling and editing light interactions in digital content creation is important for redirecting attention or setting the mood of an overall scene. Recently, several methods have been proposed to provide animators with tools to control the appearance of light [KPD10, NJS*11, SNM*13, GRR*16, NDVZJ19], see Schmidt et al. [SPN*16] for a recent survey. Among light interactions, caustics are interesting and visually appealing phenomena that create beautiful and complex patterns of light and color. These patterns are produced when light rays are reflected or refracted from specular material surfaces, producing sharp structures and details that are challenging to simulate. However, artistically controlling dynamic caustics patterns through fine-tuned constraints is not possible given current methods. Thus, the aim of this work is to provide an intuitive and precise tool to create artist-directed caustic animations. The caustics are thereby meant to break the physical constraints of geometric optics, allowing for instance fluid-like patterns. Our method enables complex caustic movements, which could be used in marketing and for visual effects. We aim to morph patterns from an initial photon distribution to a target distribution in a smooth and controllable manner. Previously, Günther et al. [GRR*16] focused on the coherent evolution of caustic patterns in an animation, where intermediate key frames also resemble plausible caustics. However, the proposed method does not provides dynamic intuitive control over the intermediate photon paths, and scenes with time-varying photon densities may exhibit undesirable motions, which were addressed by extensive parameter tuning. In this paper, we propose a novel method to animate caustic photons to desired targets. This is accomplished by transporting photons using vector fields that can be divergence-free, irrotational or a linear combination of the two.

To compute vector fields that will dynamically transport photons, we follow three main steps. First, we trace caustic photons from an initial source distribution (Sec. 2). Afterwards, we establish a correspondence between the source and a given target distribution via a modified linear assignment problem [GRR*16]. According to the target, we compute a transport vector field once by fitting a stream function and a scalar potential through an energy minimization, which reduces to a linear least squares optimization (Sec. 3). Thereby, the co-gradient of the stream function compactly encodes the divergence-free component of the flow and the gradient of the scalar potential encodes its irrotational contribution, giving control over flow properties. Both components are artistically weighted along with constraints that ensure that the vector field is smooth, follows desired directions at certain key frames and does not cross specified boundaries. Finally, advected photon positions are progressively rendered in each frame of the resulting animation (Sec. 4). Note that the resulting animations are non-physical, since the index of refraction of the specular objects is not optimized to obtain the resulting caustic pattern.

2. Photon Tracing

Input to our method is a 3D scene that contains at least one curved specular surface (such as a glass sphere) that casts caustics onto a

diffuse surface. First, we trace photons using progressive photon mapping [HJ09]. For simplicity, we assume that photons land on a plane and can thus be processed in a 2D domain. Each photon stores its position, the RGB color flux, the last specular mesh hit by the photon and the light source that emitted it. With this, we can distinguish caustics that were produced from different light emitters and meshes, as previously suggested by Reiner et al. [RKRD12]. The photon position and flux serve as initial states in the blending.

3. Photon Animation

This step contains our novel method for the artistic blending of caustics. Formally, we define the set of 2D caustic photons on the diffuse surfaces as *source* points $\mathbf{a}_k \in \mathcal{A}$. Given a desired target image, we use importance sampling as in [GRR*16] to obtain the set of *target* points $\mathbf{b}_k \in \mathcal{B}$, with $K = |\mathcal{A}| = |\mathcal{B}|$. We compute the vector field $\mathbf{u}$ in a $M \times N$ grid by fitting a stream function $\psi(\mathbf{x},t)$ and a scalar potential $\phi(\mathbf{x},t)$ to a set of constraints. Both, $\psi(\mathbf{x},t)$ and $\phi(\mathbf{x},t)$ are thereby discretized onto a grid with user-defined resolution. Using a weighted combination of the co-gradient of ψ and the gradient of ϕ, we get:

$$\mathbf{u}(\mathbf{x},t) = (1-\gamma)\underbrace{\begin{pmatrix} -\frac{\partial\psi(\mathbf{x},t)}{\partial y} \\ \frac{\partial\psi(\mathbf{x},t)}{\partial x} \end{pmatrix}}_{\text{divergence-free}} + \gamma\underbrace{\begin{pmatrix} \frac{\partial\phi(\mathbf{x},t)}{\partial x} \\ \frac{\partial\phi(\mathbf{x},t)}{\partial y} \end{pmatrix}}_{\text{irrotational}}. \tag{1}$$

Note that the co-gradient of a stream function is always divergence-free and that a gradient field is always irrotational. The weight γ in Eq. (1) gives artistic control, as it allows the user to balance between a divergence-free solution ($\gamma = 0$), which behaves like a liquid, and an irrotational solution ($\gamma = 1$), which acts like a gradient flow.

Matching. We use the method of Günther et al. [GRR*16] to determine a bijective mapping $\sigma : \mathbb{N} \to \mathbb{N}$ that assigns each source point $\mathbf{a}_k$ a unique target point $\mathbf{b}_{\sigma(k)}$. In our method, this first assignment is used as guidance direction of each photon. All photons in $\mathcal{A}$ follow the flow created by the vector field to reach the targets in $\mathcal{B}$. We call $\mathbf{c}_k(t)$ the unknown trajectory that photon k has to follow to arrive at $\mathbf{b}_{\sigma(k)}$, so that $\mathbf{c}_k(0) = \mathbf{a}_k$ and $\mathbf{c}_k(1) = \mathbf{b}_{\sigma(k)}$. W.l.o.g., we assume that the animation of two consecutive key frames is parameterized by $t \in [0,1]$.

Guidance Direction. To obtain end-point interpolation we compute each time step a guidance direction that assumes a direct linear path and leads particles in the right direction. At each curve point $\mathbf{c}_k(t)$ of the photon path, we scale the guidance direction $\mathbf{v}_k(t)$ in which the particle has to go so that particles reach the target $\mathbf{b}_{\sigma(k)}$ exactly at time $t = 1$:

$$\mathbf{v}_k(t) = \frac{\mathbf{b}_{\sigma(k)} - \mathbf{c}_k(t)}{1-t} \tag{2}$$

To make the overall motion of all particles divergence-free, irrotational or a linear combination of the two, we fit a stream function and a scalar potential so that the resulting flow aligns at photon positions $\mathbf{c}_k(t)$ with their guidance directions $\mathbf{v}_k(t)$.

User Constraints. A main advantage of using a vector field to lead photons to a target is the possibility of adding constraints, such as *directed lines* or *wall lines*, that give the artist more control over the flow and, consequently, the path that the photons will follow:

- **Directed lines:** Let $\mathbf{d}_p \in \mathcal{D}$ be the set of points with $P = |\mathcal{D}|$ that represent a *directed line* painted by the user. The direction of the enforced flow $\mathbf{w}_p$ starts from the first point $\mathbf{d}_0$ and goes in the direction of subsequent points, i.e., from $\mathbf{d}_p$ to $\mathbf{d}_{p+1}$. To add this constraint, we enforce that at grid cells containing a directed line, the desired direction aligns with the vector field $\mathbf{u}$.
- **Wall Lines:** In order to avoid flow through certain regions, the user can draw *wall lines* across which photons should not move. To enable a flow around walls, we redirect the guidance direction (Eq. (2)) of photons close to a wall by computing the shortest path to the target. For this, we embed a transport graph in the domain that connects adjacent cells unless they are separated by a wall. The new guidance direction $\mathbf{v}'_k(t)$ is in direction of the shortest path from the current position $\mathbf{c}_k(t)$ to target $\mathbf{b}_{\sigma(k)}$, where we use the first two vertices of the shortest path, computed with Dijkstra's algorithm, to obtain the direction, and the total length of the path to scale the velocity so that the photon will reach the target exactly at $t = 1$. Computing the shortest path for each photon is costly and thus we only compute the path for photons inside or close to the convex hull of a wall. Thereby, the *thickness* of the wall is a user parameter, which we set by default to three grid cells.

Vector Field Optimization. We compute the vector field $\mathbf{u}$ as a linear combination of the co-gradient of a stream function ψ and the gradient of a scalar potential ϕ, cf. Eq. (1). We used the previously introduced ingredients to formulate the energy that we minimize to obtain stream function ψ and scalar potential ϕ fields for each time step t:

$$E = \frac{1}{2K}\sum_k^K \left\| (1-\gamma)\begin{pmatrix} -\frac{\partial\psi(\mathbf{c}_k(t))}{\partial y} \\ \frac{\partial\psi(\mathbf{c}_k(t))}{\partial x} \end{pmatrix} + \gamma\begin{pmatrix} \frac{\partial\phi(\mathbf{c}_k(t))}{\partial x} \\ \frac{\partial\phi(\mathbf{c}_k(t))}{\partial y} \end{pmatrix} - \mathbf{v}_k^{[t]}(t) \right\|^2 \tag{3}$$

$$+ \frac{\alpha(1-\gamma)}{2|D|}\int_D \left(\frac{\partial^2\psi}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2\psi}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2\psi}{\partial y^2}\right)^2 \mathrm{d}\mathbf{x} \tag{4}$$

$$+ \frac{\alpha\gamma}{2|D|}\int_D \left(\frac{\partial^2\phi}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2\phi}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2\phi}{\partial y^2}\right)^2 \mathrm{d}\mathbf{x} \tag{5}$$

$$+ \frac{\beta}{2P}\sum_p^P \left\| (1-\gamma)\begin{pmatrix} -\frac{\partial\psi(\mathbf{d}_p)}{\partial y} \\ \frac{\partial\psi(\mathbf{d}_p)}{\partial x} \end{pmatrix} + \gamma\begin{pmatrix} \frac{\partial\phi(\mathbf{d}_p)}{\partial x} \\ \frac{\partial\phi(\mathbf{d}_p)}{\partial y} \end{pmatrix} - \mathbf{w}_p \right\|^2 \tag{6}$$

First, Term (3) ensures that the guidance direction $\mathbf{v}_k(t)$ is set for each photon when constructing the flow using Eq. (1). In the convex hull of walls, we use $\mathbf{v}'_k(t)$ instead. Terms (4) and (5) enforce smoothness by minimizing the bending energy using thin-plate splines [Boo89]. Term (6) adds directed lines, i.e., curves that the user wishes the flow to follow. The discretization of the quadratic energy $\widehat{E}$ can be found in the additional material. By setting the gradient with respect to the unknowns to zero via $\frac{\partial\widehat{E}(\psi,\phi,t)}{\partial\psi} = \frac{\partial\widehat{E}(\psi,\phi,t)}{\partial\phi} = 0$, the energy $\widehat{E}$ is minimized. This results in a set of linear equations, which are known as the *normal equations*.

In our case, we get:

$$\left(\frac{1}{K}\mathbf{A}^{\mathrm{T}}\mathbf{A}+\frac{\alpha}{MN}\mathbf{B}^{\mathrm{T}}\mathbf{B}+\frac{\beta}{P}\mathbf{D}^{\mathrm{T}}\mathbf{D}\right)\begin{pmatrix}\psi\\ \phi\end{pmatrix}=\frac{1}{K}\mathbf{A}^{\mathrm{T}}\mathbf{a}+\frac{\beta}{P}\mathbf{D}^{\mathrm{T}}\mathbf{d} \quad (7)$$

which is a $2MN \times 2MN$ system that can be linearly solved for ψ and ϕ. We employ outflow conditions for exterior boundary velocities, which implies applying Neumann boundary conditions to the linear system of Eq. (7). Note that since all constraints are defined by derivatives of ψ and ϕ, the solution is unique up to an added constant. We set one component of ψ and ϕ to zero to select one solution. In the special cases of $\gamma = 0$ and $\gamma = 1$, the linear problem can be directly solved for either ψ or ϕ, which simplifies to a $MN \times MN$ system.

Advection and Flux Blending. We numerically advect all particles in $\mathbf{u}$ to the next time step using Eq. (1). We use a fourth-order Runge-Kutta (RK4) integrator to model their trajectories $\mathbf{c}(t)$ as tangent curves of the vector field, i.e., $\frac{\mathrm{d}\mathbf{c}(t)}{\mathrm{d}t} = \mathbf{u}(\mathbf{c}(t),t)$. When only using vector field $\mathbf{u}(\mathbf{x},t)$ it is generally not possible to exactly reach the desired target positions, since the resolution of ψ and ϕ is limited. Thus, we use the guidance direction $\mathbf{v}_k^{[\prime]}(t)$ from Eq. (2) to linearly fade at the end of the animation:

$$\frac{\mathrm{d}\mathbf{c}(t)}{\mathrm{d}t} = (1-s)\cdot\mathbf{u}(\mathbf{c}(t),t)+s\cdot\mathbf{v}_k^{[\prime]}(t)\ ,\ \text{with}\ s=\begin{cases}0 & t\le\tau\\ \dfrac{t-\tau}{1-\tau} & \tau<t\end{cases}$$

where $\tau \in [0, 1]$ is the time at which the blending coefficient s starts to increase, which we set to $\tau = 0.85$. When the remaining distance to the target is below a certain threshold, we exclude the photon from the minimization and linearly blend the remaining path to the target.

Each caustic photon $\mathbf{a}_k \in \mathcal{A}$ has a flux $\phi_{\mathbf{a}_k}$. Since we importance sample the target positions $\mathbf{b}_k \in \mathcal{B}$ according to the target image brightness, we can assume that all target locations have the same flux $\phi_{\mathbf{b}}$. We distribute the total source flux uniformly across the target locations by computing its average: $\phi_{\mathbf{b}} = \frac{\rho}{|\mathcal{A}|}\sum_{\mathbf{a}_k\in\mathcal{A}}\phi_{\mathbf{a}_k}$,where ρ is a user parameter that allows us to scale the brightness of the target image. Note that for $\rho = 1$, we obtain energy preservation, which is the default setting. At time t, the resulting flux of a photon is a linear blend $\phi_{\mathbf{c}_{k(t)}} = (1-t)\cdot\phi_{\mathbf{a}_k} + t\cdot\phi_{\mathbf{b}}$.

4. Photon Rendering

Finally, we use stochastic progressive photon mapping [HJ09] to render the photons. The rendering is split over multiple iterations, each processing a separate batch of photons. For the mapping from source to target, we reuse the continuous bijective map constructed from the first set of photons using the method of Günther et al. [GRR*16]. To ensure convergence, the kernel density estimation radius reduces in each iteration [KZ11]. Since we render the background image without caustics in advance, the user can use a small caustic photon map to obtain a fast preview, which proved helpful in the creation process.

5. Results

We tested our method with different scenes and target distributions and refer to the video for the resulting animations. In the additional material, we performed a parameter study (grid resolution, blending weight γ), and list the time spend on photon tracing (Step 1), blending of photons with our method (Step 2) and finally rendering 100 frames of the animation (Step 3) for our test scenes. Step 1 is in the order of $1-2$ minutes, whereas Step 2 and Step 3 are in the order of $3-11$ hours in our prototype implementation.

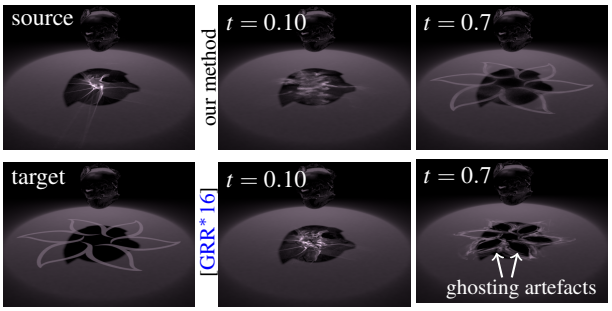


Figure 1: *Comparison of direct blending to a drawing with our method (top row) and previous work by Günther et al. [GRR*16] (bottom row) in the* DARK GLASS *scene. In their approach, caustics can move on top of each other, which creates ghosting artefacts.*

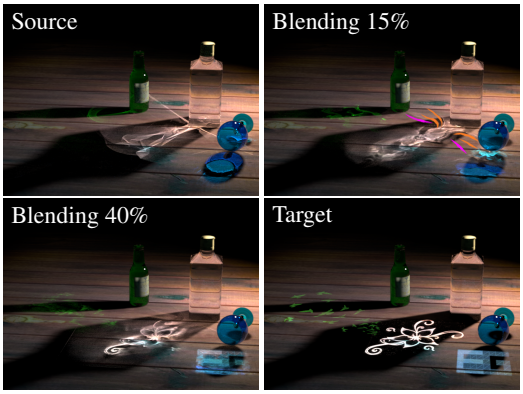
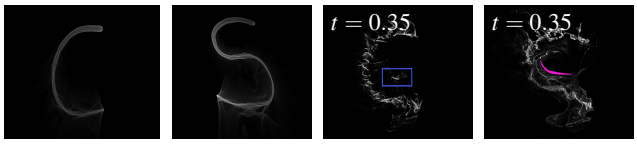


Figure 2: *Smooth animation of three caustics morphed into different target images in the* BOTTLES *scene. Walls (magenta, top right) prevent mixing of caustics, while directed lines (orange, top right) guide caustics towards specified targets.*

Comparison with Previous Work In Fig. 1, we blend a caustic into a flower and compare our animation with the method of Günther et al. [GRR*16]. Their method uses a weighted blending between cubic B-splines and linear paths, whereas our method uses a vector field to transport the photons. We can observe that, although the behavior is very similar when no constraints are drawn, previous work has noticeable ghosting artifacts in the intermediate frames. This is produced by groups of photons coming from different locations to the same target, which eventually looks like images moving slowly on top of each other. Since we use a smooth (potentially divergence-free) vector field, photons paths rarely cross and thus our method does not exhibit this artifact.

User Constraints The use of *directed lines* and *wall lines* in our method greatly adds artistic freedom to the animations and prevents undesired movements in the animation. Fig. 2 shows an example of the use of walls and directed lines to redirect the flow. Next to



(a) Source (b) Target (c) [GRR*16] (d) Our method

Figure 3: *Example of a letter C (a) that is blended to letter S (b). In previous work (c), parts of the caustic break off (see blue box). With our wall line (magenta) in (d), this artefact is avoided.*

the white caustic, we placed two walls (visualized in magenta) to avoid the white straight line of light to be mixed with the green and blue caustics in the animation. This level of control was not available in previous work [GRR*16]. When using only the walls, the flow takes photons closer to the walls until they get redirected around them by the path planning. The addition of directed lines (visualized in orange) helps to channel the movement of the fluid through the opening between the walls, resulting in a smoother movement towards the center. Fig. 3 shows another example with an undesired splitting of photons when blending an image of a *C* letter to an *S* letter. Using [GRR*16], a group of photons (in blue) splits off and moves upwards to a different arc of the letter, which required tedious and indirect parameter adjustment. With our method, the easy placement of a wall (in magenta) forces photons to travel along the letter.

Synthetic Turbulence. As a post-process, we allow users to add synthetic turbulence to the flow **u**. To construct a turbulent vector field, Bridson et al. [BHN07] computed the curl of a scalar field (which is divergence-free) that was generated by Perlin noise. In 2D, the scalar field is a stream function and the curl is the co-gradient. Using this, we add multiple synthetic turbulence scales to our vector field. A *turbulence* value scales the amount of randomness we add when creating random noise vector fields and interpolating at each frame between them. This generates either a fast random flow (high *turbulence*) or a slower and less strong flow (a low *turbulence*). The grid used to generate the random vector field influences the size of the vortices that the noise field produces. More advanced methods use style transfer [SDKN18] or wavelet turbulence [KTJG08], which directly creates vortices at certain scales with a proper energy-related weighting. Fig. 4 shows an example where we used our turbulence model to disturb the photon paths.

6. Conclusions

In this paper, we proposed a method to animate caustics into a given target distribution by describing the photon trajectories as particles moving in a controllable vector field. This way, vector field design tools become available, which allow us to introduce new constraints that help users to direct the flow, create boundaries or add additional effects such as synthetic turbulence. With the newly introduced constraints, we no longer need tedious manual parameter adjustments to avoid undesired movement of photons, and we can explore the usage of synthetic turbulence. The proposed method can still be improved so that the flow computed is efficiently focused in the regions where the amount of photons is bigger. Further, we currently assumed planar receiving surfaces. Reprojection [GRR*16] or on-surface deformation [RTD*10] could be used to lift this limitation.

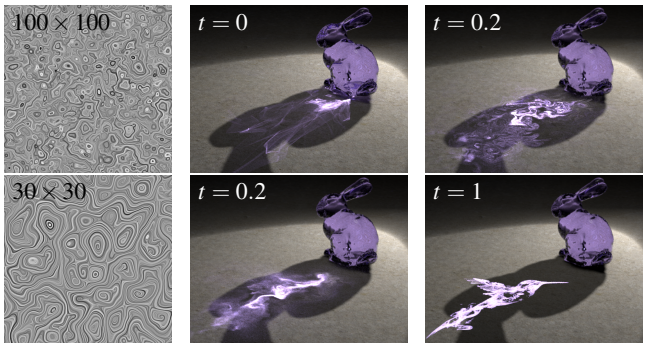


Figure 4: *Addition of turbulent vector fields with turbulence* = 20 *in the* BUNNY *scene. Top: the caustic received the* 100×100 *noise. Bottom:* 30×30 *noise. We refer to the video for an animation.*

References

[BHN07] BRIDSON R., HOURIHAM J., NORDENSTAM M.: Curl-noise for procedural fluid flow. *ACM Trans. Graph. (Proc. SIGGRAPH) 26* (2007), 46. 4

[Boo89] BOOKSTEIN F. L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence 11*, 6 (1989), 567–585. 2

[GRR*16] GÜNTHER T., ROHMER K., RÖSSL C., GROSCH T., THEISEL H.: Stylized caustics: Progressive rendering of animated caustics. *Comp. Graph. Forum (Proc. EG) 35* (2016), 243–252. 1, 2, 3, 4

[HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 28*, 5 (2009), 141:1–141:8. 2, 3

[KPD10] KERR W. B., PELLACINI F., DENNING J. D.: Bendylights: Artistic control of direct illumination by curving light rays. *Computer Graphics Forum (Proc. EGSR) 29*, 4 (2010), 1451–1459. 1

[KTJG08] KIM T., THÜREY N., JAMES D., GROSS M.: Wavelet turbulence for fluid simulation. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 50:1–50:6. 4

[KZ11] KNAUS C., ZWICKER M.: Progressive photon mapping: A probabilistic approach. *ACM Trans. Graph. 30*, 3 (2011), 25:1–25:13. 3

[NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia) 38*, 6 (Nov. 2019). 1

[NJS*11] NOWROUZEZAHRAI D., JOHNSON J., SELLE A., LACEWELL D., KASCHALK M., JAROSZ W.: A programmable system for artistic volumetric lighting. *ACM Transactions on Graphics 30*, 4 (July 2011), 29:1–29:8. 1

[RKRD12] REINER T., KAPLANYAN A., REINHARD M., DACHSBACHER C.: Selective inspection and interactive visualization of light transport in virtual scenes. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 711–718. 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. 4

[SDKN18] SATO S., DOBASHI Y., KIM T., NISHITA T.: Example-based turbulence style transfer. *ACM Trans. Graph. 37*, 4 (2018). 4

[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

[SPN*16] 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 35* (2016), 216–233. 1