# Controllable Caustic Animation Using Vector Fields Additional Material

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

## 1. Discretization

We discretize the stream function $\psi_{i,j}$ and the scalar potential $\phi_{i,j}$ onto a regular $M \times N$ grid. The energy from Eqs. (9)–(12) of the main paper is then discretized as:

$$\widehat{E} = \frac{1}{2K}\left(\mathbf{A}\begin{pmatrix}\psi\\ \phi\end{pmatrix} - \mathbf{a}\right)^2 + \frac{\alpha}{2MN}\left(\mathbf{B}\begin{pmatrix}\psi\\ \phi\end{pmatrix}\right)^2 + \frac{\beta}{2P}\left(\mathbf{D}\begin{pmatrix}\psi\\ \phi\end{pmatrix} - \mathbf{d}\right)^2.$$

Using the gradient $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^{\mathrm{T}}$ and co-gradient $\nabla_\perp = (-\frac{\partial}{\partial y}, \frac{\partial}{\partial x})^{\mathrm{T}}$:

$$\mathbf{A}\begin{pmatrix}\psi\\ \phi\end{pmatrix} = \begin{bmatrix}\vdots\\ (1-\gamma)\nabla_\perp\psi(\mathbf{c}_k(t)) + \gamma\nabla\phi(\mathbf{c}_k(t))\\ \vdots\end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix}\vdots\\ \mathbf{v}_k^{[l]}(t)\\ \vdots\end{bmatrix}, \tag{1}$$

$$\mathbf{B}\begin{pmatrix}\psi\\ \phi\end{pmatrix} = \Big[\, (1-\gamma)\frac{\partial^2\psi^{\mathrm{T}}}{\partial x^2} \quad (1-\gamma)\frac{\sqrt{2}\partial^2\psi^{\mathrm{T}}}{\partial x \partial y} \quad (1-\gamma)\frac{\partial^2\psi^{\mathrm{T}}}{\partial y^2} \quad \gamma\frac{\partial^2\phi^{\mathrm{T}}}{\partial x^2} \quad \gamma\frac{\sqrt{2}\partial^2\phi^{\mathrm{T}}}{\partial x \partial y} \quad \gamma\frac{\partial^2\phi^{\mathrm{T}}}{\partial y^2}\,\Big]^{\mathrm{T}}, \tag{2}$$

$$\mathbf{D}\begin{pmatrix}\psi\\ \phi\end{pmatrix} = \begin{bmatrix}\vdots\\ (1-\gamma)\nabla_\perp\psi(\mathbf{d}_p) + \gamma\nabla\phi(\mathbf{d}_p)\\ \vdots\end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix}\vdots\\ \mathbf{w}_p\\ \vdots\end{bmatrix}, \tag{3}$$

where $\psi$ and $\phi$ are $MN \times 1$ vectors that contain the stream function scalars and the scalar potential scalars of the entire domain. $\mathbf{A}$ is a $2K \times 2MN$ matrix, $\mathbf{a}$ is a $2K \times 1$ vector, $\mathbf{B}$ is a $6MN \times 2MN$ matrix, $\mathbf{D}$ is a $2P \times 2MN$ matrix and $\mathbf{d}$ is a $2P \times 1$ vector.

## 2. Algorithm Overview

A pseudocode overview of our algorithm is listed in Alg. 1. Each progressive iteration, $K$ photons are traced, for which a matching with the target distribution is computed using [GRR*16]. In each frame of the animation, a vector field is computed and the photons are advected. Finally, the photons are added to the image and rendered using progressive photon mapping.

```
Input: directed lines w_p, target image I_target
Output: images I of animation
for it ← 1 to numProgressiveIterations do
    // Step 1: Photon tracing
    a_k, φ_{a_k} ← tracePhotons(K)
    b_k ← sampleTarget(I_target, K)

    // Step 2: Photon animation
    σ(k) ← computeMatching(a_k, b_k)   // Sec. 4.3
    c_k ← a_k
    for frame ← 1 to numFrames do
        v_k^[l] ← computeGuidance(b_σ(k), c_k)   // Sec. 4.4, 4.5
        φ, ψ ← computeFlow(c_k, w_p, v_k^[l])   // Sec. 4.6
        u ← (1 − γ) ∇⊥φ + γ∇ψ                   // Eq. (4)
        c_k, φ_{c_k} ←advectPhotons(c_k, u, v_k^[l], φ_{a_k})  // Sec. 4.8

        // Step 3: Photon rendering
        I_frame ← ppm(c_k, φ_{c_k})
    end
end
```

**Algorithm 1:** Algorithm overview.

| Scene | Step 1 | Step 2 | Step 3 |
|---|---|---|---|
| BOTTLES (6M) | 2 min | 6 h | 11 h |
| DARK GLASS (4M) | 1 min | 5 h | 3.5 h |
| BUNNY (4M) | 1 min | 9.7 h | 3 h |

**Table 1:** *Timings in 3 scenes for the caustic tracing (Step 1), the blending (Step 2) and the rendering of 100 frames for the final animations (Step 3). The latter uses 4-6 million caustic photons, 35 progressive iterations and a stream function grid $\psi_{i,j}$ of $400 \times 400$.*

### 2.1. Performance

The setup of the linear system is in $\mathcal{O}(K+p+n)$ and its solution is in $\mathcal{O}(n^3)$, with $K$ being the number of photons, $p$ being the number of directed line constraints and $n$ being the number of grid points. Table 1 shows the details of the computational time for our test scenes in this paper. For all examples, we used an Intel® Core™i5-4570 CPU @ 3.20GHz processor with 16.0 GB of memory RAM.

### 2.2. Parameters

**Balance between Divergence-Free and Irrotational Flow.** Fig. 1 compares the results obtained for different constant $\gamma$. We refer to the video for an animation of this parameter, demonstrating that

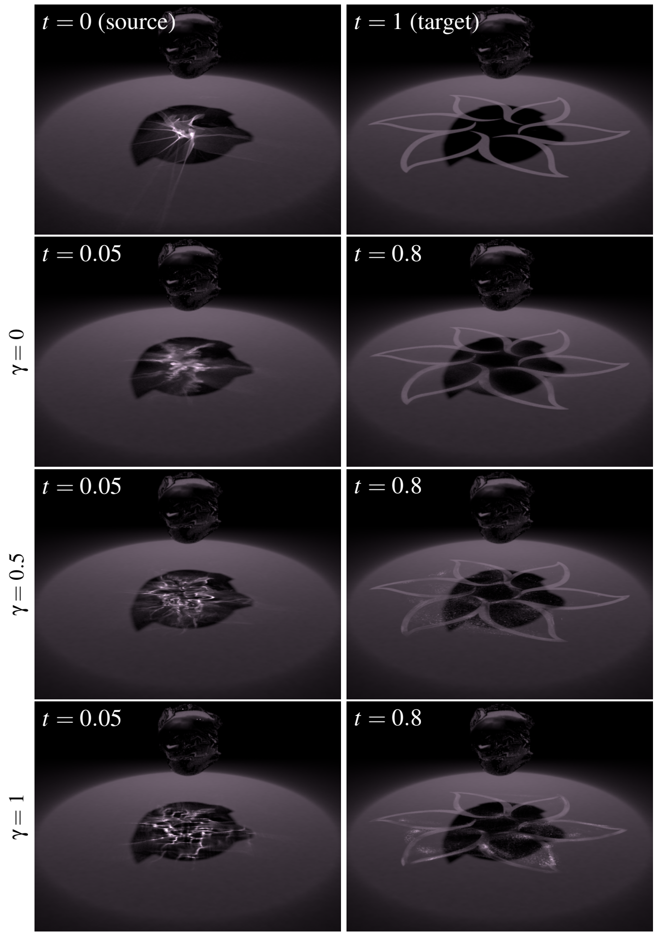


**Figure 1:** *Parameter study for a constant $\gamma$, balancing between a divergence-free solution ($\gamma = 0$) and an irrotational solution ($\gamma = 1$).*

the solution blends continuously when varying $\gamma$. When setting $\gamma$ too high, the flow contains sinks in which photons cluster, which becomes a visible artefact. If setting $\gamma$ too low, particles take a long time to collapse onto a smaller target due to the incompressibility of the vector field. In Fig. 2, we show a test with a time-dependent $\gamma(t)$ that starts with a divergence-free flow and gradually increases the compressibility towards the end. This helps photons to collapse onto the target, since a divergent-free flow avoids this behaviour and thus is not desired towards the end of the animation.

**Grid Resolution.** In the main paper, we solve for the stream function $\psi$ and the scalar potential $\phi$ on a discrete regular $M \times N$ grid. For all our examples, we used $400 \times 400$. In Fig. 3, we show results for different resolutions. The higher the resolution, the more fidelity can be represented by the vector field and thus photons move closer to their target.

**Linear Path Threshold.** Fig. 4 shows an example for different thresholds, where early switch to linear paths results in smoother animations, whereas late switching (smaller threshold) can result in fast movements at the end of the animation. We refer to video for animations of this parameter.

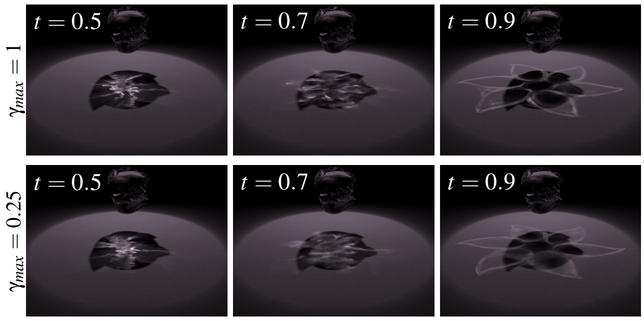


**Figure 2:** *Varying $\gamma$ over time with $\gamma(t) = \gamma_{max} \cdot (2t^3 - 3t^2)$ allows us to start with a divergence-free flow which slowly transitions towards an irrotational flow with a certain maximal irrotational component $\gamma_{max}$. If the irrotational component is too strong (top row), photon clustering may create visible artifacts.*

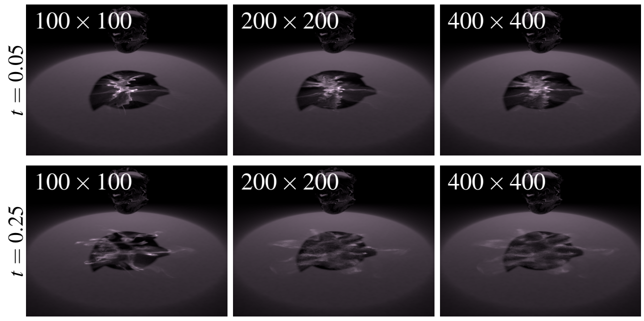


**Figure 3:** *Comparison of different grid resolutions of the stream function discretization $\psi_{i,j}$ and scalar potential discretization $\phi_{i,j}$. The higher the resolution, the closer are fine structures transported to their actual target.*

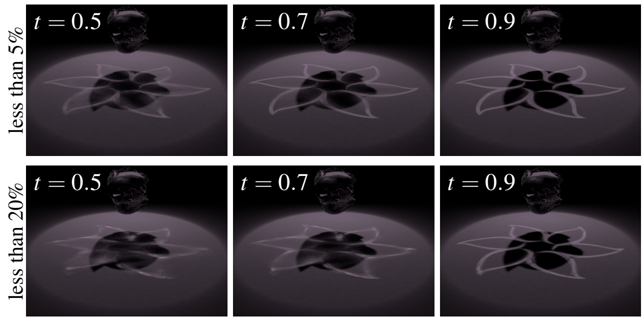


**Figure 4:** *When photons get closer to their target than a certain percentage of their total path, we switch to a linear blending and let them no longer participate in the energy minimization. The earlier we switch, the less blurry are the final frames (see bottom row).*

## References

[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