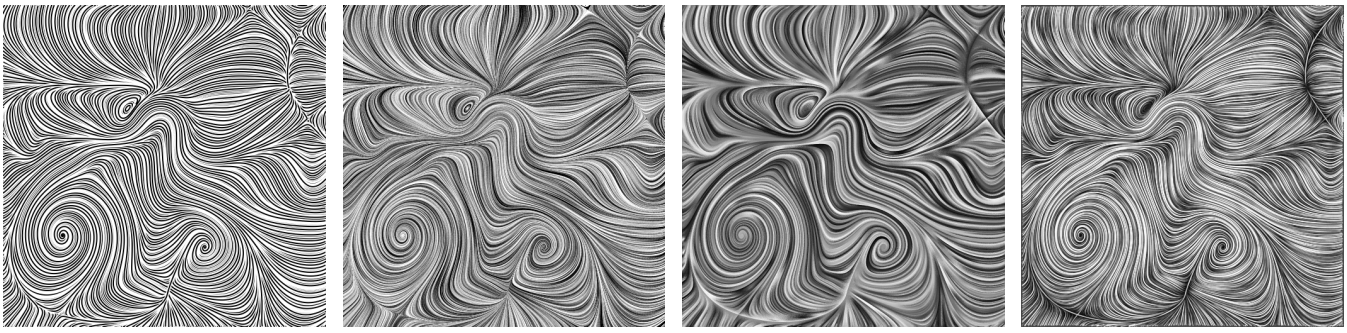
Monte Carlo Methods for 2D Flow Visualization

Xingze Tian and Tobias Günther

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany



Evenly-spaced streamlines | Line integral convolution | Smooth vector graphics (ours) | Light transport simulation (ours)

Figure 1: *In this paper, we explore visually aesthetic alternatives to common steady 2D vector field visualization techniques. From left to right, evenly-spaced streamlines [JL97], line integral convolution [CL93], as well as our two proposed approaches are shown.*

Abstract

In this paper, we investigate how recent advances from the computer graphics literature can be applied to improve the visualization of two-dimensional vector fields. To this end, we propose two different approaches that both start from a set of evenly-spaced streamlines. The first approach avoids the need for contrast normalization, which is usually required for LIC approaches. For this, the image synthesis is phrased as a diffusion problem by placing double-sided Dirichlet boundary conditions along the streamlines. The diffusion problem is formally modeled as a linear elliptic partial differential equation, which is solved stochastically using a variant of the walk-on-spheres algorithm in order to achieve anti-aliased results. The second approach leverages human's perception of shape to convey flow patterns. For this, we lift the streamlines into the third dimension and generate visual contrast among adjacent streamlines by means of ambient occlusion. To synthesize the images, we apply a physically based material model and employ a Monte Carlo renderer to simulate the light transport.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization;*** • ***Computing methodologies*** → *Rendering;*

1. Introduction

Flow visualization is concerned with the visual analysis of vector fields. For two-dimensional data sets, it is common practice to start the exploration with the well-established line integral convolution (LIC) [CL93] or by placing a dense set of streamlines [JL97]. Even for three-dimensional domains, a common starting point is to explore slices using these approaches. Although there have been extensions in the following years, the core concepts of the algorithms were established in the 1990s, around the time OpenGL was just released by Silicon Graphics (1992). Much progress has been made in the graphics community over the past 30 years. When polishing 3D visualizations for publications in prestigious journals or for science communication, the utilization of global illumination, ambient occlusion, or area lights has become a common choice for scientific visualizations [MSRT13, GKT16, DG24], resulting in eye catching images. There is, however, a striking difference in aesthetics when it comes to 2D flow visualization, where techniques such as LIC are still the standard practice. Using currently trending techniques from computer graphics [NDVZJ19, SC20], we propose two 2D vector field visualizations that primarily aim to be visually pleasant, while at the same time not sacrificing too much on the effectiveness. Both visualizations start from a set of evenly-spaced streamlines [JL97], which are faded out at the endpoints in different ways. Our first approach formulates the image synthesis as a dif-

fusion process by utilizing diffusion curves [OBW*08]. Thereby, each streamline receives a color as Dirichlet boundary condition on the left and right side, and the color of each pixel is then computed by a diffusion process. For a subpixel-accurate and anti-aliased rendering, we solve the diffusion problem stochastically using the Walk-on-Spheres (WoS) algorithm [Mul56, SC20], which is currently extended and actively researched in the geometry processing community [SMGC23, SCJ*23, MSCG24]. To implement fading at the endpoints of curves, we extend the WoS algorithm to handle transmissive boundary conditions by lifting the random walks into an additional dimension that allows walks over the boundaries. Unlike LIC, this approach does not require a contrast enhancing post-process. The second approach adds visual contrast among adjacent streamlines by calculating ambient occlusion through a light transport simulation [NDVZJ19]. To this end, the line geometry is extruded into the third dimension and the width of the geometry is reduced towards the endpoints of the streamlines. Unlike LIC, the darkening of lines is grounded in the shadowing that humans are naturally accustomed to, aiming for more pleasant visualizations. In summary, we make the following contributions:

- We propose two 2D flow visualization methods that aim for a visually pleasing communication of flow patterns.
- We extend the Walk-on-Spheres algorithm to handle transmissive boundary conditions, for which we formulate the underlying PDE with probabilistic boundary conditions.

2. Related Work

Among various flow visualization techniques, one common and effective group is the group of texture-based methods, which aim to create a dense representation of the vector field using an underlying texture [LHD*04]. The texture can be generated with spot noise [VW91], where each spot uses a uniform intensity function to represent a particle moving slightly in time, creating a streak that follows the direction of the local flow. Later, spot noise was applied to experimental flow visualization [dLPPW95], and further enhanced techniques such as parallel spot noise [QSWW10] and unsteady spot noise [LJH03] have also been proposed. Although spot noise methods can convey the velocity magnitude of the flow, they are less effective in representing the flow direction and critical points than LIC techniques [dLvL98], which are developed from the line integral convolution algorithm [CL93]. Later, fast LIC was introduced to speed up the computation time of LIC by minimizing streamline computation time and reusing some of the convolution results [SH95]. The range of LIC has also been extended from 2D regular grids to non-linear grids [For94], unsteady flow [FC95, SK97] and 3D flow [IG97, IG98, RSHTE99]. However, due to the nature of how LIC images are constructed, the results are often blurry and noisy with low contrast, making feature detection difficult [OK97, VKP99]. Okada et al. [OK97] proposed enhanced LIC, which applies the LIC algorithm twice and uses two filters for clearer lines as a post-processing process. In addition, they used the velocity direction to color the texture image for better feature detection in the final result. Hege and Stalling used a higher order filter kernel for smoother LIC results on any surfaces [HS98]. Verma et al. [VKP99] leveraged the advantages of both streamline visualizations and LIC techniques.

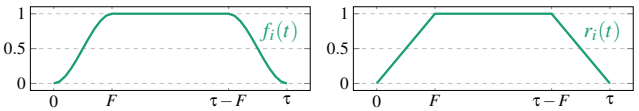


Figure 2: *The fading function $f_i(t)$ is used to fade out a streamline with arclength τ over a distance F at the beginning and ending of the line. The fading function $f_i(t)$ is computed by applying a* smoothstep *to the piece-wise linear function $r_i(t)$.*

3. Method

The goal of this paper is to explore visually pleasant alternatives to the common line integral convolution (LIC) [CL93]. For this, we propose two novel flow visualization approaches that are based on Monte Carlo methods used in smooth vector graphics [TG24] and light transport simulation [Jar08]. In the following, we will first explain the common input to both approaches.

3.1. Streamline Generation

Given is a continuous two-dimensional steady vector field $\mathbf{v}(\mathbf{x}) : \mathcal{D} \to \mathbb{R}^2$, which is defined in the spatial domain $\mathcal{D} \subset \mathbb{R}^2$. The starting point for both our approaches is the generation of a dense set $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ of N streamlines $\mathbf{s}_i(t)$ that cover the domain $\mathcal{D}$. Each streamline $\mathbf{s}(t)$ is uniquely determined by a seed point $\mathbf{s}(t_0) = \mathbf{s}_0$, and we require all streamlines to be tangential to the *normalized* vector field $\mathbf{v}(\mathbf{x})$:

$$\frac{\mathrm{d}\mathbf{s}(t)}{\mathrm{d}t} = \frac{\mathbf{v}(\mathbf{s}(t))}{\|\mathbf{v}(\mathbf{s}(t))\|} \quad \text{s.t.} \quad \mathbf{s}(t_0) = \mathbf{s}_0. \tag{1}$$

Formally, we define a streamline over the domain $\mathbf{s}_i(t) : [0, \tau_i] \to \mathcal{D}$, where τ_i is the total arclength of the streamline $\mathbf{s}_i$. We define a fading function $f_i(t) : [0, \tau_i] \to [0, 1]$ per streamline $\mathbf{s}_i$, which is zero at the endpoints and one in the interior of the curve domain:

$$f_i(t) = 3r_i(t)^2 - 2r_i(t)^3, \quad r_i(t) = \min\left\{1, \frac{t}{F}, \frac{\tau_i - t}{F}\right\}. \tag{2}$$

Thereby, F denotes the distance over which the fading takes place. Fig. 2 illustrates the fading function. The fading is not applied to endpoints on the domain boundary $\partial\mathcal{D}$ to avoid unnecessary fading at the image border, and is reduced when the density of streamline endpoints is high to show critical points. To represent saddle critical points well, we add separatrices [GBR21] to the streamline set $\mathcal{S}$. The remaining space in the domain is filled by applying the evenly-spaced streamline seeding algorithm of Jobard and Lefer [JL97]. The calculation of distance to existing lines is accelerated with kd-trees. The streamlines are numerically integrated using the fourth-order Runge-Kutta method (RK4).

3.2. Smooth Vector Graphics

Conventional LIC [CL93] requires a post-process to enhance the contrast, since the convolution with uniform noise converges to the average gray value. When the streamlines have different lengths, for example, because some streamlines exit the domain early, contrast must be adjusted locally per streamline. Instead, we propose to treat streamlines as diffusion curves [OBW*08], i.e., a random color is defined on the left and right side of each streamline, which

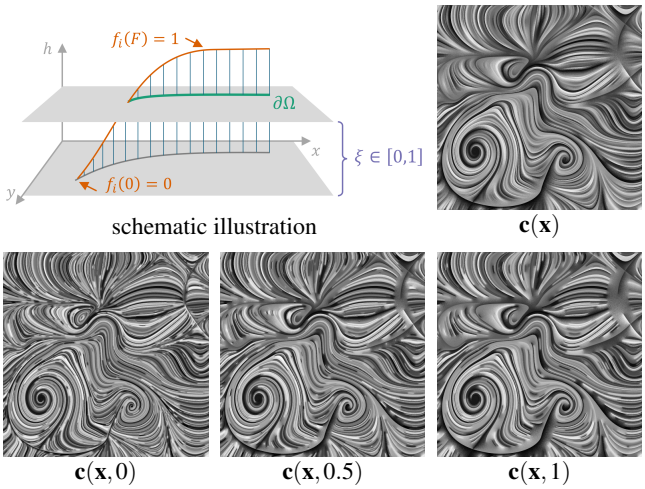


Figure 3: *Schematic illustration of the transmissive Dirichlet conditions. Consider the fading function $f_i(t)$, which determines the 'height' of the Dirichlet boundaries. For illustrative purposes, we rendered individual horizontal slices in the bottom row. The final image $\mathbf{c}(\mathbf{x})$ is the average of the horizontal slices.*

is diffused into the domain. While we keep the notation general and refer to colors on the left and right side, we only use gray values in this paper, since we intentionally leave the color channel open for the encoding of other attributes.

We follow Jeschke et al. [JCW09] and model diffusion curves as double-sided boundaries with Dirichlet conditions. Formally, we divide the domain $\mathcal{D}$ into the interior $\Omega \in \mathbb{R}^2$ and its boundary $\partial\Omega$. The boundary $\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$ consists of the image boundary $\partial\Omega_N$ and the two sides of the streamlines themselves $\partial\Omega_D$:

$$\partial\Omega_D = \{\mathbf{s}_i^+, \mathbf{s}_i^- \mid \mathbf{s}_i \in \mathcal{S}\}. \tag{3}$$

The two sides of each streamline $\mathbf{s}_i$ have opposite normal orientation $\mathbf{n}_i^+ = -\mathbf{n}_i^-$ and separate colors $\mathbf{g}_i^+, \mathbf{g}_i^-$, which are randomly sampled from a uniform distribution. In previous work on smooth vector graphics, boundary curves have always been modeled as opaque, not allowing colors to leak through. Instead, we model *transmissive* Dirichlet boundary conditions to achieve smooth fadings of the line endings, as shown in Fig. 3. The bottom row shows images with non-transmissive boundaries, which are non-smooth at line endings. The fading term in Eq. (2) varies along the curve $f(\mathbf{x}) := f_i(t)$ for $\mathbf{x} = \mathbf{c}_i(t)$, which we conceptually model as the 'height' of the curve. Thus, the curve is low at the end points and high in the middle. The higher the curve, the less color leaks through, which results in smoother transitions at the endpoints. To determine the final pixel color $\mathbf{c}(\mathbf{x}) : \Omega \to \mathcal{C}$ with $\mathcal{C} \subset \mathbb{R}^3$, we perform Monte Carlo integration over the height range $[0,1]$ by uniformly sampling $h \in [0,1]$ and treating the boundaries as closed only when $h < f_i(t)$:

$$\mathbf{c}(\mathbf{x}) = \int_0^1 \mathbf{c}(\mathbf{x},h)\, \partial h \tag{4}$$

For a given h, we are interested in the color field $\mathbf{c}(\mathbf{x},h) : \Omega \to \mathcal{C}$

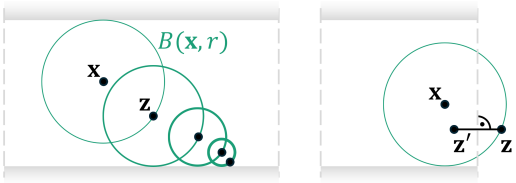


Figure 4: *Illustration of the Walk-on-Spheres. Left: The walk continues recursively until a Dirichlet condition is reached. Right: At homogeneous Neumann boundary conditions points are reflected.*

that fulfills the Laplace equations:

$$\Delta\mathbf{c}(\mathbf{x},h) = \mathbf{0} \qquad \forall \mathbf{x} \in \Omega \tag{5}$$

$$\mathbf{c}(\mathbf{x},h) = \mathbf{g}(\mathbf{x}) \qquad \forall \mathbf{x} \in \partial\Omega_D \wedge h < f(\mathbf{x}) \tag{6}$$

$$\frac{\partial \mathbf{c}(\mathbf{x},h)}{\partial \mathbf{n}_\mathbf{x}} = \mathbf{0} \qquad \forall \mathbf{x} \in \partial\Omega_N \tag{7}$$

where $\mathbf{g}(\mathbf{x})$ is the respective color from $\{\mathbf{g}_i^+, \mathbf{g}_i^-\}$ of the streamline $\mathbf{s}_i$ containing $\mathbf{x} \in \mathbf{s}_i$. Eq. (5) implies that the color field $\mathbf{c}(\mathbf{x},h)$ is spatially smooth in the interior of the domain. Eq. (6) applies the Dirichlet boundary condition along the diffusion curves only if h is below the fading term, which lowers the probability of the existence of the boundary as the line fades out. Eq. (7) applies a homogeneous Neumann boundary condition to the border of the image.

To compute $\mathbf{c}(\mathbf{x},h)$, we use the Walk-on-Spheres (WoS) [Mul56, SC20], which spans the largest possible ball $\mathcal{C} = \mathcal{B}(\mathbf{x},r)$ around $\mathbf{x}$ that does not intersect any Dirichlet boundary. Then, the color $\mathbf{c}(\mathbf{x},h)$ is integrated over the boundary of the ball $\mathbf{z} \in \mathcal{B}(\mathbf{x},r)$:

$$\mathbf{c}(\mathbf{x},h) = \frac{1}{|\mathcal{B}(\mathbf{x},r)|} \int_{\partial\mathcal{B}(\mathbf{x},r)} \mathbf{c}(\mathbf{z},h)\, \mathrm{d}\mathbf{z}, \tag{8}$$

The idea of WoS is to compute the unknown color $\mathbf{c}(\mathbf{z},h)$ recursively by generating another ball around $\mathbf{z}$, as illustrated in Fig. 4 (left). In essence, a random walk on recursively generated spheres (or balls) is performed, and it terminates when the walk gets sufficiently close to a diffusion curve. When this happens, the Dirichlet condition in Eq. (6) provides the color for the endpoint of the walk. The size of the next sphere is determined by finding the distance to the closest boundary. At the beginning of each walk, we draw a uniform random number $\xi \in [0,1]$ to determine the height h in Eq. (4), which decides the boundary $\partial\Omega$, cf. Fig. 3. With Eq. (7), we place homogeneous Neumann boundary conditions on the image boundary. For a random walk, this means that walks that exit the domain are simply reflected back into the domain [Gre07], as shown in Fig. 4 (right). We implemented our approach on the GPU using compute shaders. Unlike with LIC, the colors placed by the Dirichlet boundary conditions in Eq. (6) retain a high contrast.

3.3. Light Transport Simulation

The smooth vector graphics formulation lends its aesthetic appeal from the smooth color gradients between lines. This entails, however, that the streamline density is not as high as with LIC. Thus, we further propose another approach, which starts from a denser set of lines. A conventional LIC visualization exhibits bright and dark lines that are aligned with the flow. Which lines are darkened is

random. In our human perception, darkening is a visual cue related to shape perception, in particular indentations exhibit a darkening due to ambient occlusion. Thus, with this paper, we aim to create a three-dimensional structure that exhibits indentations aligning with the line structures. Similar to LIC, this results in bright and dark line structures that align with the flow, but human perception can make better sense of the visual cues. The calculation of ambient occlusion in the indentations of the three-dimensional geometry requires a physically-based light transport simulation.

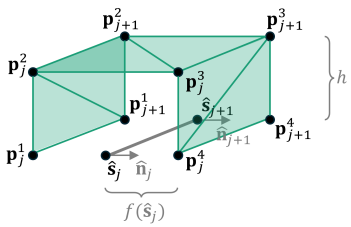
In a physically-based renderer, the intensity of a pixel depends on the scene extent. To unify the light setup across all scenes, we initially rescale the scene to fit the streamlines into the unit domain $[-1,1]^2$. The input streamlines are 1-dimensional lines embedded in a 2-dimensional domain. For rendering in a light transport simulation, we extrude the lines to 2-dimensional surfaces embedded in a 3-dimensional domain. Since we aim for a varying amount of ambient occlusion between adjacent lines, we elevate each streamline $\mathbf{s}$ to an empirically chosen random height $h \in [0, 0.1]$, With this, each streamline vertex gives rise to four points:

$$\mathbf{p}_j^1 = \begin{pmatrix} \hat{\mathbf{s}}_j - \hat{\mathbf{n}}_j \cdot f(\hat{\mathbf{s}}_j) \\ 0 \end{pmatrix}, \mathbf{p}_j^2 = \begin{pmatrix} \hat{\mathbf{s}}_j - \hat{\mathbf{n}}_j \cdot f(\hat{\mathbf{s}}_j) \\ h \end{pmatrix}, \tag{9}$$

$$\mathbf{p}_j^3 = \begin{pmatrix} \hat{\mathbf{s}}_j + \hat{\mathbf{n}}_j \cdot f(\hat{\mathbf{s}}_j) \\ h \end{pmatrix}, \mathbf{p}_j^4 = \begin{pmatrix} \hat{\mathbf{s}}_j + \hat{\mathbf{n}}_j \cdot f(\hat{\mathbf{s}}_j) \\ 0 \end{pmatrix}, \tag{10}$$

which are connected with the next vertex to form a triangle strip:

$$\left(\mathbf{p}_j^1, \mathbf{p}_{j+1}^1, \mathbf{p}_j^2, \mathbf{p}_{j+1}^2, \mathbf{p}_j^3, \mathbf{p}_{j+1}^3, \mathbf{p}_j^4, \mathbf{p}_{j+1}^4\right). \tag{11}$$



To avoid unwanted perspective distortion, we apply an orthographic projection from the top. The camera is centered above the scene to have exact pixel correspondence between grid coordinates and pixel coordinates. The scene is illuminated with a uniform area light source that extends 30% out of the horizontal bounding box and hovers 1 unit above the ground. The light source emits a radiance of 3 units of power per unit area per unit steradian. For the microfacet model, an isotropic GGX distribution [WMLT07] with a roughness of 0.2 is used. The extruded line geometry is placed on a ground plane, using a Lambertian BSDF with a reflectance of 0.3.

The light transport simulation solves the radiative transport equation [Jar08], which expresses the exitant radiance $L(\mathbf{x} \to \omega)$ at $\mathbf{x}$ going out in direction ω:

$$L(\mathbf{x} \to \omega) = \underbrace{L_e(\mathbf{x} \to \omega)}_{\text{emitted}} + \underbrace{\int_H f_r(\mathbf{x}, \omega' \leftrightarrow \omega)\, L(\mathbf{x} \leftarrow \omega')\, (\mathbf{n} \cdot \omega') \partial\omega'}_{\text{reflected}}.$$

Thereby, $L_e(\mathbf{x} \to \omega)$ expresses the radiance that is emitted at $\mathbf{x}$ in direction ω, for example by the area light source. The term $f_r(\mathbf{x}, \omega' \leftrightarrow \omega)$ describes the BSDF, which accounts for the fraction of light that is reflected at $\mathbf{x}$ from the direction ω' towards the direction ω. The in-scattered $L(\mathbf{x} \leftarrow \omega)$ is calculated recursively and expresses the radiance arriving at $\mathbf{x}$ from direction ω'. Lastly, the dot product between surface normal $\mathbf{n}$ and incident direction ω' accounts for foreshortening. The integral $\int_H$ collects the incident light over the visible hemisphere. Numerically, the recursive integral is Monte Carlo estimated using a path tracer [Kaj86], for which we used Mitsuba [NDVZJ19] through its Python binding.

4. Results

We test both our approaches on slices of different vector fields. Fig. 1 compared conventional evenly-spaced streamlines [JL97] and contrast-enhanced line integral convolution [CL93] with our two proposed methods using the BORRO data set by [CB11]. We refer to the supplemental material for similar comparisons in other scenes. Both our approaches are Monte Carlo methods, i.e., their solution is calculated iteratively, starting from a noisy solution. The supplemental material contains a convergence series, showing how the noise vanishes over time. Further, the supplemental material contains performance measurements for both approaches on all test scenes. The smooth vector graphics rendering takes 116-275 seconds for 100 spp (samples per pixel), depending on the scene. The time for the light transport simulation remained consistently at about 75 seconds for 1,024 spp. Due to the non-Lambertian BSDF, the light transport simulations takes more spp to reach a converged image. Lastly, we apply three different contrast enhancement methods to all four methods, showing that their effect is negligible on our methods, while it is well-known to be required for LIC [HS98].

5. Conclusions

In this paper, we revisited a problem that nowadays many would consider solved: the visualization of steady 2D vector fields. With the aim to develop visual encodings that are more amenable for science communication, we transferred recent approaches from the computer graphics literature into the field of flow visualization. To utilize a smooth vector graphics representation of the flow by means of diffusion curves, we introduced a transmissive Walk-on-Spheres formulation that allows rendering diffusion curves with translucent Dirichlet boundary conditions. Further, we achieved a darkening of neighboring lines using ambient occlusion by extruding the geometry into the third dimension. For the necessary light transport simulation, we used a state-of-the-art path tracer. Compared to the line integral convolution, both our approaches need more time to obtain converged results. The utility of our approaches resides therefore less in interactive exploration scenarios, but rather in presentation tasks. The Walk-on-Spheres solver could be accelerated with boundary value caching [MSCG23]. Further, a grid-based diffusion curve solver [OBW*08] is an alternative, where special care is needed to anti-alias the result. Further, transmissive Dirichlet boundary conditions have not been developed for grid-based approaches, although for this it would be possible to discretize the integral in Eq. (4) with a Riemannian sum.

References

[CB11] CANDELARESI S., BRANDENBURG A.: Decay of helical and nonhelical magnetic knots. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics 84*, 1 (2011), 016406. doi:10.1103/PhysRevE.84.016406. 4

[CL93] CABRAL B., LEEDOM L. C.: Imaging vector fields using line integral convolution. In *Proceedings of the 20th Annual Conference*

on Computer Graphics and Interactive Techniques (New York, NY, USA, 1993), SIGGRAPH '93, Association for Computing Machinery, p. 263–270. doi:10.1145/166117.166151. 1, 2, 4

[DG24] DASSLER N., GÜNTHER T.: Variational feature extraction in scientific visualization. *ACM Transactions on Graphics (TOG) 43*, 4 (2024), 1–16. doi:10.1145/3658219. 1

[dLPPW95] DE LEEUW W. C., PAGENDARM H.-G., POST F. H., WALTER B.: Visual simulation of experimental oil-flow visualization by spot noise images from numerical flow simulation. In *Visualization in Scientific Computing* (1995), Springer, pp. 135–148. doi:10.1007/978-3-7091-9425-6_13. 2

[dLvL98] DE LEEUW W., VAN LIERE R.: Comparing LIC and spot noise. In *Proceedings Visualization '98* (1998), pp. 359–365. doi:10.1109/VISUAL.1998.745324. 2

[FC95] FORSSELL L. K., COHEN S. D.: Using line integral convolution for flow visualization: Curvilinear grids, variable-speed animation, and unsteady flows. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 133–141. doi:10.1109/2945.468406. 2

[For94] FORSSELL L. K.: Visualizing flow over curvilinear grid surfaces using line integral convolution. In *Proceedings Visualization'94* (1994), IEEE, pp. 240–247. doi:10.1109/VISUAL.1994.346313. 2

[GBR21] GÜNTHER T., BAEZA ROJO I.: Introduction to vector field topology. In *Topological Methods in Data Analysis and Visualization VI* (Cham, 2021), Springer International Publishing, pp. 289–326. doi:10.1007/978-3-030-83500-2_15. 2

[GKT16] GÜNTHER T., KUHN A., THEISEL H.: MCFTLE: Monte Carlo rendering of finite-time Lyapunov exponent fields. *Computer Graphics Forum (Eurographics Conference on Visualization) 35*, 3 (2016), 381–390. doi:10.1111/cgf.12914. 1

[Gre07] GREBENKOV D. S.: NMR survey of reflected brownian motion. *Reviews of Modern Physics 79*, 3 (2007), 1077–1137. doi:10.1103/RevModPhys.79.1077. 3

[HS98] HEGE H.-C., STALLING D.: Fast LIC with piecewise polynomial filter kernels. *Mathematical Visualization: Algorithms, Applications and Numerics* (1998), 295–314. doi:10.1007/978-3-662-03567-2_22. 2, 4

[IG97] INTERRANTE V., GROSCH C.: *Strategies for effectively visualizing 3D flow with volume LIC*. IEEE, 1997. doi:10.1109/VISUAL.1997.663912. 2

[IG98] INTERRANTE V., GROSCH C.: Visualizing 3D flow. *IEEE Computer Graphics and Applications 18*, 4 (1998), 49–53. doi:10.1109/38.689664. 2

[Jar08] JAROSZ W.: *Efficient Monte Carlo methods for light transport in scattering media*. PhD thesis, University of California, San Diego, 2008. 2, 4

[JCW09] JESCHKE S., CLINE D., WONKA P.: A GPU Laplacian solver for diffusion curves and Poisson image editing. *ACM Transactions on Graphics 28*, 5 (2009), 1–8. doi:10.1145/1618452.1618462. 3

[JL97] JOBARD B., LEFER W.: Creating evenly-spaced streamlines of arbitrary density. In *Visualization in Scientific Computing '97* (Vienna, 1997), Lefer W., Grave M., (Eds.), Springer Vienna, pp. 43–55. doi:10.1007/978-3-7091-6876-9_5. 1, 2, 4

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (1986), pp. 143–150. doi:10.1145/15922.15902. 4

[LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum* (2004). doi:10.1111/j.1467-8659.2004.00753.x. 2

[LJH03] LARAMEE R. S., JOBARD B., HAUSER H.: Image space based visualization of unsteady flow on surfaces. In *IEEE Visualization, 2003. VIS 2003.* (2003), IEEE, pp. 131–138. doi:10.1109/VISUAL.2003.1250364. 2

[MSCG23] MILLER B., SAWHNEY R., CRANE K., GKIOULEKAS I.: Boundary value caching for walk on spheres. *ACM Trans. Graph. 42*, 4 (jul 2023). doi:10.1145/3592400. 4

[MSCG24] MILLER B., SAWHNEY R., CRANE K., GKIOULEKAS I.: Walkin' Robin: Walk on stars with robin boundary conditions. *ACM Trans. Graph. 43*, 4 (jul 2024). URL: https://doi.org/10.1145/3658153, doi:10.1145/3658153. 2

[MSRT13] MARTINEZ ESTURO J., SCHULZE M., RÖSSL C., THEISEL H.: Global selection of stream surfaces. *Computer Graphics Forum (Proc. Eurographics) 32*, 2 (2013), 113–122. doi:10.1111/cgf.12031. 1

[Mul56] MULLER M. E.: Some continuous Monte Carlo methods for the Dirichlet problem. *The Annals of Mathematical Statistics 27*, 3 (1956), 569 – 589. doi:10.1214/aoms/1177728169. 2, 3

[NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics 38*, 6 (2019), 1–17. doi:10.1145/3355089.3356498. 1, 2, 4

[OBW*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion curves: a vector representation for smooth-shaded images. *ACM Trans. Graph. (TOG) 27*, 3 (2008), 1–8. doi:10.1145/2483852.2483873. 2, 4

[OK97] OKADA A., KAO D. L.: Enhanced line integral convolution with flow feature detection. In *Visual Data Exploration and Analysis IV* (1997), vol. 3017, SPIE, pp. 206–217. 2

[QSWW10] QIN B., SU F., WU Z., WANG J.: GPU based spot noise parallel algorithm for 2D vector field visualization. In *2010 International Conference on Optoelectronics and Image Processing* (2010), vol. 1, IEEE, pp. 580–583. doi:10.1109/ICOIP.2010.292. 2

[RSHTE99] REZK-SALAMA C., HASTREITER P., TEITZEL C., ERTL T.: Interactive exploration of volume line integral convolution based on 3D-texture mapping. In *Proceedings Visualization'99* (1999), IEEE, pp. 233–528. doi:10.1109/VISUAL.1999.809892. 2

[SC20] SAWHNEY R., CRANE K.: Monte Carlo geometry processing: A grid-free approach to PDE-based methods on volumetric domains. *ACM Trans. Graph. 39*, 4 (jul 2020). doi:10.1145/3386569.3392374. 1, 2, 3

[SCJ*23] SUGIMOTO R., CHEN T., JIANG Y., BATTY C., HACHISUKA T.: A practical walk-on-boundary method for boundary value problems. *ACM Trans. Graph. 42*, 4 (jul 2023). doi:10.1145/3592109. 2

[SH95] STALLING D., HEGE H.-C.: Fast and resolution independent line integral convolution. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), pp. 249–256. doi:10.1145/218380.218448. 2

[SK97] SHEN H.-W., KAO D. L.: UFLIC: a line integral convolution algorithm for visualizing unsteady flows. In *Proceedings. Visualization'97 (Cat. No. 97CB36155)* (1997), IEEE, pp. 317–322. doi:10.1109/VISUAL.1997.663898. 2

[SMGC23] SAWHNEY R., MILLER B., GKIOULEKAS I., CRANE K.: Walk on stars: A grid-free Monte Carlo method for PDEs with Neumann boundary conditions. *ACM Trans. Graph. 42*, 4 (jul 2023). doi:10.1145/3592398. 2

[TG24] TIAN X., GÜNTHER T.: A survey of smooth vector graphics: Recent advances in representation, creation, rasterization and image vectorization. *IEEE Transactions on Visualization and Computer Graphics 30*, 3 (2024), 1652–1671. doi:10.1109/TVCG.2022.3220575. 2

[VKP99] VERMA V., KAO D., PANG A.: PLIC: Bridging the gap between streamlines and LIC. In *Proceedings Visualization'99 (Cat. No. 99CB37067)* (1999), IEEE, pp. 341–541. 2

[VW91] VAN WIJK J. J.: Spot noise texture synthesis for data visualization. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (1991), pp. 309–318. 2

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. *Rendering techniques 2007* (2007), 18th. 4