Stroke Transfer for Participating Media

NAOTO SHIRASHIMA*, Aoyama Gakuin University (AGU), Japan
HIDEKI TODO*, Takushoku University, Japan
YUKI YAMAOKA, Aoyama Gakuin University (AGU), Japan
SHIZUO KAJI, Kyushu University, Japan, Kyoto University, Japan
KUNIHIKO KOBAYASHI, Aoyama Gakuin University (AGU), Japan
HARUNA SHIMOTAHIRA, Aoyama Gakuin University (AGU), Japan
YONGHAO YUE, Aoyama Gakuin University (AGU), Japan

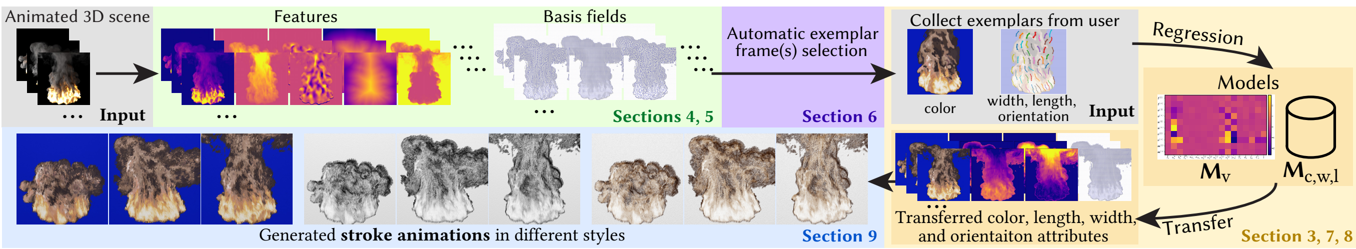


Fig. 1. **Pipeline of our method**. Starting with a 3D scene containing participating media (top left), our method generates animated painterly drawings (bottom left). The pipeline begins by computing features and basis fields from the 3D scene, followed by the automatic selection of a minimal set of exemplar frames that best represent the feature distribution across the animation. Users provide exemplars with the four key stroke attributes—color, width, length, and orientation—required for synthesizing strokes. Using these exemplars, we perform regression to train models $\mathbf{M}_\mathrm{c,w,l}$ and $\mathbf{M}_\mathrm{v}$, enabling the transfer of stroke attributes to animation frames and the creation of stylistically varied painterly strokes.

We present a method for generating stroke-based painterly drawings of participating media, such as smoke, fire, and clouds, by transferring stroke attributes—color, width, length, and orientation—from exemplar to animation frames. Building on the stroke transfer framework, we introduce features and basis fields capturing lighting-, view-, and geometry-dependent information, extending surface-based ones (e.g., intensity, apparent normals and curvatures, and distance from silhouettes) to volumetric scenes while supporting traditional surface objects. Novel features, including apparent relative velocity and mean free-path, address non-rigid flow and dynamic scenes. Our system combines automated exemplar selection, user-guided style learning, and temporally coherent stroke generation, enabling artistic and expressive visualizations of dynamic media.

CCS Concepts: • **Computing methodologies → Non-photorealistic rendering**.

*Joint first authors — authors contributed equally.

Authors' Contact Information: Naoto Shirashima, Aoyama Gakuin University (AGU), Kanagawa, Japan, koutei7penguin.jp@gmail.com; Hideki Todo, Takushoku University, Tokyo, Japan, htodo@cs.takushoku-u.ac.jp; Yuki Yamaoka, Aoyama Gakuin University (AGU), Kanagawa, Japan, ymok124@gmail.com; Shizuo Kaji, Kyushu University, Fukuoka, Japan, and Kyoto University, Kyoto, Japan, skaji@imi.kyushu-u.ac.jp; Kunihiko Kobayashi, Aoyama Gakuin University (AGU), Kanagawa, Japan, kuni.koba.onepiece@outlook.com; Haruna Shimotahira, Aoyama Gakuin University (AGU), Kanagawa, Japan, sanso.sugu@gmail.com; Yonghao Yue, Aoyama Gakuin University (AGU), Kanagawa, Japan, yonghao@it.aoyama.ac.jp.

SIGGRAPH Conference Papers '25, Vancouver, BC, Canada

ACM ISBN 979-8-4007-1540-2/2025/08
https://doi.org/10.1145/3721238.3730603

Additional Key Words and Phrases: Non-photorealistic rendering, stroke-based rendering, example-based, stroke transfer, vector field generation, participating media, volumetric normals and curvatures, automatic exemplar selection

ACM Reference Format:
Naoto Shirashima, Hideki Todo, Yuki Yamaoka, Shizuo Kaji, Kunihiko Kobayashi, Haruna Shimotahira, and Yonghao Yue. 2025. Stroke Transfer for Participating Media. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3721238.3730603

1 Introduction

Participating media—such as smoke, fog, fire, and clouds—exhibit rich spatial and temporal complexity, ranging from smooth laminar flows to turbulent swirls, and from wispy transparencies to dense glowing volumes. While these phenomena have been extensively studied in photorealistic rendering [Cerezo et al. 2005; Novák et al. 2018] and fluid simulation [Bridson 2008; Koschier et al. 2022], their non-photorealistic rendering (NPR) remains comparatively unexplored, despite its potential for artistic and expressive visualization.

We present a stroke-based method for generating painterly animations of dynamic 3D scenes containing participating media. NPR for such scenes poses unique challenges due to *volumetric occlusion*, where multiple 3D points may project onto the same pixel, complicating the design of spatial features for stylization [Durand 2002; Schmid et al. 2011].

Our approach adopts a *single-layer model*, compositing opaque or semi-transparent strokes in screen space to represent volumes and surfaces in a unified way. Although painterly NPR spans many

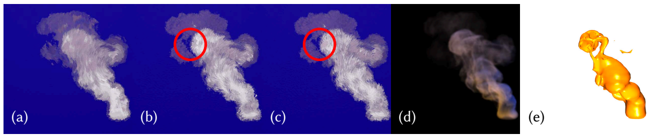


Fig. 2. **Comparison using alternative features.** Stroke transfer results using (a) our features, (b) image-based features (luminance, color, intensity gradient from the rendered image (d) and motion vectors from optical flow), and (c) isosurface-based features (luminance, color, intensity gradient from (d) and geometric features from the isosurface (e)). Image-based and isosurface-based features often fail to handle subtle wisps correctly (resulting in, e.g., unnatural gaps highlighted by the red circles). Additionally, noisy motion vector estimation in the image-based approach causes temporal stroke noise, while abrupt topological changes in the isosurface destabilize normal and curvature computations, leading to popping artifacts (see the accompanying videos for details).

stylistic dimensions [Hegde et al. 2013; Hertzmann 2003], we focus on four core stroke attributes: color, width, length, and orientation.

Building upon the stroke transfer framework of Todo et al. [2022], we compute scalar feature fields and basis fields from the 3D scene at each frame. We extend this framework by formulating *volumetric features* that naturally incorporate surface features as a limiting case. The computed fields are mapped to stroke attributes via simple regressors trained on a small set of user-annotated exemplar frames, favoring *interpretability* over black-box models. To further minimize user effort, we introduce an *automated exemplar selection* algorithm. No external datasets or pre-trained networks are required.

Alternative strategies for defining volumetric features include extracting isosurfaces and applying surface-based methods. However, mismatches between fuzzy volumetric distributions and sharp surface boundaries often produce unnatural gaps (Figure 2), and small changes in density can trigger abrupt topological transitions, destabilizing curvature computation. Another possibility is to extract image-space features from rendered volumes, such as optical flow [Jamriška et al. 2019]; however, inaccuracies in flow estimation introduce noise, and important volumetric cues like temperature are lost, leading to flattened appearance and distorted color transitions.

Motivated by these limitations, we compute volumetric features directly from 3D fields. Our design leverages the connection between surface and volumetric visibility: surface visibility corresponds to a *delta function*, while volumetric media are described by a smooth *free-path distribution*. As extinction increases, the distribution converges to a delta function, enabling a principled generalization from surfaces to volumes. We also incorporate cues such as velocity and temperature to better capture dynamic and emissive effects.

Rather than handcrafting a minimal set of features and basis fields, we construct a broad candidate pool and perform data-driven adaptive selection. Ablation studies show this produces more expressive and temporally coherent results.

Our method preserves stroke structure more reliably than patch-based approaches (e.g.,[Texler et al. 2020]) and neural style transfer methods (e.g.,[Hu et al. 2023]), which often suffer from artifacts or inconsistent shapes. Compared to surface-limited stroke transfer [Todo et al. 2022], our method supports both surfaces and volumetric media, improves temporal coherence, and reduces manual intervention through automated exemplar selection. We demonstrate the versatility of our method across a range of scenarios, including hybrid surface-volume scenes, fire, and turbulent flows.

2 Related Work

2.1 Painterly rendering

Since the seminal work by Haeberli [1990], numerous methods have been developed to transform images or 3D scenes into painterly art forms (e.g., [Kalnins et al. 2003; Meier 1996]). These have been extended to ensure temporal coherence across animation frames [Hays and Essa 2004; Hertzmann and Perlin 2000; Kalnins et al. 2003]. Digital painting tools enable artwork creation on 2D digital canvases (e.g., [Baxter et al. 2001; Chen et al. 2015]) or 3D volumetric canvases (e.g., [Katanics and Lappa 2003]), with later advancements supporting painting directly within volumes [Schmid et al. 2011]. Our method focuses on transforming dynamic 3D scenes with participating media into painterly animations. For broader overviews, see the surveys of Hertzmann [2003] and Hegde et al. [2013].

2.2 NPR techniques for participating media

In addition to image-space techniques for photorealistic rendering, explicit methods have been developed for handling participating media, including fluid simulation techniques like morphing via keyframes [Browning et al. 2014] and flow-guided texture synthesis [Jamriška et al. 2015]. Stylized rendering approaches explore cartoon-style effects [Álvarez et al. 2007; Eden et al. 2007; McGuire and Fein 2006; Selle et al. 2004] and volumetric textures via neural style transfer [Guo et al. 2021; Kim et al. 2019, 2020]. Artist-directed effects can also be achieved with photon beam techniques [Nowrouzezahrai et al. 2011]. However, these methods do not explicitly model brushstrokes, a key focus of our work.

2.3 Example-based painterly animation

Example-based painterly techniques often rely on patch-based synthesis [Bénard et al. 2013; Fišer et al. 2016; Hertzmann et al. 2001; Jamriška et al. 2019; Platkevič et al. 2021; Sýkora et al. 2019], as reviewed in [Barnes and Zhang 2017]. Neural methods generate stylized images [Futschik et al. 2021; Gatys et al. 2016; Ghiasi et al. 2017; Liu et al. 2023; Susladkar et al. 2024; Texler et al. 2020], with some addressing temporal coherence [Yang et al. 2023]. However, these methods often require large datasets and lack interpretability or fine stroke-level control.

Stroke-based approaches [Cardona and Saito 2015; Haga et al. 2001; Kalnins et al. 2003; Kalogerakis et al. 2012; Kowalski et al. 1999; Lee et al. 2010; Markosian et al. 2000; Northrup and Markosian 2000; Olsen et al. 2005; Singh and Schaefer 2010; Yan et al. 2008] instead construct compositions from explicit strokes and generally avoid the stitching artifacts of patch-based methods. Recent neural techniques adopt stroke-aware representations [Hu et al. 2023; Kotovenko et al. 2021; Liu et al. 2021, 2023], though many still struggle with temporal coherence. For a comprehensive discussion, see the survey of Nolte et al. [2022].

The stroke transfer framework of Todo et al. [2022] combines stroke composition with interpretable learning of stroke attributes.

Our method generalizes this framework to support volumetric effects and dynamic behavior of participating media.

2.4 Diffusion-based text-to-video

Recent diffusion-based methods synthesize videos from text prompts [Chefer et al. 2024; Guo et al. 2024; Kondratyuk et al. 2024; Liu et al. 2024; Wang et al. 2023], and commercial platforms, such as Runway Gen2 and HailuoAI, offer user-friendly access to these capabilities. These models emphasize realism and semantic alignment and may allow conditioning on input sequences (e.g., depth or segmentation). However, they lack explicit, stroke-level modeling of spatio-temporal structure. Our method focuses on artistic, stroke-based rendering of participating media, enabling detailed control over stroke attributes (e.g., orientation, color, length, width) and coherence through physically informed feature fields. We rely on a small number of user-provided exemplars and structured screen-space features derived from 3D input, instead of large pretrained models.

2.5 Photorealistic rendering of participating media

Photorealistic rendering of volumetric phenomena has been extensively studied; see surveys by Cerezo et al. [2005] and Novák et al. [2018]. In exponentially distributed media, transmittance follows the Bouguer–Beer–Lambert law [Beer 1852; Bouguer 1729; Lambert 1760], and visibility is described by the free-path distribution [Yue et al. 2010]. The radiative transfer equation [Chandrasekhar 1950] underpins most rendering approaches [Miller et al. 2019; Pauly et al. 2000]. Recent works have relaxed the assumption of independent particle distributions [Bitterli et al. 2018; Jarabo et al. 2018], though our approach remains grounded in the exponential case. We adapt these physical insights to derive scalar features and basis fields for stylized volumetric rendering.

3 Overview

Our method generates stroke-based painterly drawings for participating media by transferring four key stroke attributes—color ($\boldsymbol{c}$), width (w), length (l), and orientation ($\boldsymbol{d}$)—from exemplar frames to animation frames, building upon the stroke transfer framework [Todo et al. 2022] (see Figure 1 for an illustration). For $\boldsymbol{c}$, w, and l, we define a mapping $\mathbf{M}_{\mathrm{c,w,l}}$ based on features $\boldsymbol{\phi}(t, \boldsymbol{u})$ computed at each pixel $\boldsymbol{u}$ and time t:

$$(\boldsymbol{c}(t,\boldsymbol{u}), w(t,\boldsymbol{u}), l(t,\boldsymbol{u})) = \mathbf{M}_{\mathrm{c,w,l}}(\boldsymbol{\phi}(t,\boldsymbol{u}); \Theta_{\mathrm{c,w,l}}), \tag{1}$$

with $\Theta_{\mathrm{c,w,l}}$ learned via nearest-neighbor regression for color and linear regression for width and length on the exemplar frames.

For $\boldsymbol{d}(t, \boldsymbol{u})$, we compute coefficients $\boldsymbol{a}(t, \boldsymbol{u})$ to combine basis fields $\mathbf{A}(t, \boldsymbol{u})$:

$$\boldsymbol{a}(t,\boldsymbol{u}) = \mathbf{M}_{\mathrm{v}}(\boldsymbol{\phi}(t,\boldsymbol{u}); \Theta_{\mathrm{v}}), \qquad \boldsymbol{d}(t,\boldsymbol{u}) = \mathbf{A}(t,\boldsymbol{u})\boldsymbol{a}(t,\boldsymbol{u}), \tag{2}$$

using linear regression to learn Θ_{v}.

Features and basis fields encode lighting-, view-, and geometry-dependent information observed in 2D screen space, mimicking how human artists interpret scenes. While these are intrinsic to the scene, the learned models $\mathbf{M}_{\mathrm{c,w,l}}$ and $\mathbf{M}_{\mathrm{v}}$ are style-specific and reusable across frames and scenes.

We extend feature and basis field computations to volumetric objects, ensuring compatibility with surface-only scenes. Details of these computations are presented in §4 and §5.

The system first computes features and basis fields for the input 3D scene. It then automatically identifies a minimal set of exemplar frames (§6) from which the user provides annotations for learning $\mathbf{M}_{\mathrm{c,w,l}}$ and $\mathbf{M}_{\mathrm{v}}$ (§7). Users are encouraged to supply smooth and consistent attributes for better randomness control. The attributes are transferred to animation frames using a relocation mechanism to introduce controlled randomness (§8). Finally, strokes are synthesized with enhanced temporal coherence (§9).

4 Representation of a Participating Medium

The interaction of light with participating media, governed by the radiative transport theory [Chandrasekhar 1950], defines their appearance and informs the features we compute. This section reviews key components: exponential decay, scattering, self-emission, and the behavior of the free-path distribution at high extinction.

4.1 Exponential decay

The extinction coefficient $\sigma(\boldsymbol{x})$, representing the inverse mean free path at $\boldsymbol{x}$, governs light attenuation. The optical depth, $\tau(\boldsymbol{x}_1, \boldsymbol{x}_2) = \int_{\boldsymbol{x}_1}^{\boldsymbol{x}_2} \sigma(\boldsymbol{x}) d\boldsymbol{x}$, quantifies accumulated extinction, while the transmittance, $T(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\tau(\boldsymbol{x}_1, \boldsymbol{x}_2))$, gives the probability of unimpeded travel. The free-path distribution $p_{\mathrm{fp}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma(\boldsymbol{x}_2) T(\boldsymbol{x}_1, \boldsymbol{x}_2)$ describes the likelihood of the first scattering at a specific location. The cumulative distribution function (CDF) $F_{\mathrm{fp}}(\boldsymbol{x}_1, \boldsymbol{x}_1 + \zeta\boldsymbol{\omega})$ of p_{fp} describes the probability that the first scattering occurs within a distance ζ along a direction $\boldsymbol{\omega}$ from $\boldsymbol{x}_1$:

$$F_{\mathrm{fp}}(\boldsymbol{x}_1, \boldsymbol{x}_1 + \zeta\boldsymbol{\omega}) = 1 - T(\boldsymbol{x}_1, \boldsymbol{x}_1 + \zeta\boldsymbol{\omega}). \tag{3}$$

4.2 Scattering

Scattering redistributes light upon interaction with particles. The proportion of scattering relative to the total interaction (scattering + absorption) is described by the *single scattering albedo*, $\alpha(\boldsymbol{x})$, while the *phase function* $\rho_{\omega_o}(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ models the angular scattering distribution. Combined, they are analogous to the surface BRDF. Scattering and absorption coefficients, $\sigma_s = \sigma\alpha$ and $\sigma_a = \sigma(1 - \alpha)$, sum to the total extinction coefficient σ.

4.3 Self mission

High-temperature media emit light via black-body radiation. For example, soot emits light in fire. Planck's law [Planck 1901] defines radiance $B_\nu(C) = \frac{2h\nu^3}{{c_l}^2} \frac{1}{e^{h\nu/k_B C} - 1}$ at temperature C at a frequency ν, where h is Planck's constant, c_l is the speed of light, and k_B is Boltzmann's constant.

4.4 Free-path distribution at high extinction (density) limit

At the high extinction (or equivalently, high density) limit, the free-path distribution p_{fp} approaches a delta function centered at the vacuum/non-vacuum interface. This property allows us to extend surface-defined features to volumetric media, as discussed in §5.

Consider a field $\sigma(\boldsymbol{x})$ that includes both vacuum regions ($\sigma(\boldsymbol{x}) = 0$) and non-vacuum regions ($\sigma(\boldsymbol{x}) > 0$). Suppose that we scale the

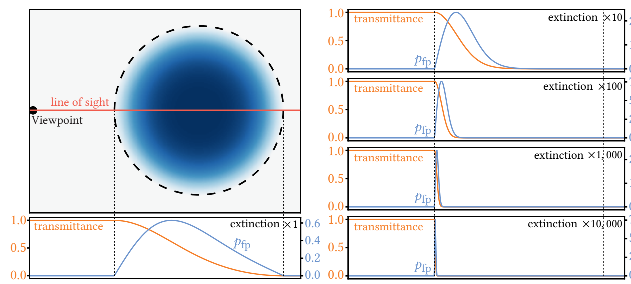


Fig. 3. **Free-path distribution at high extinction limit**. The top-left panel shows a 2D cross-section of a 3D extinction field, with extinction strictly zero outside the dashed circle. Along the red line of sight, the bottom-left panel illustrates the corresponding transmittance and free-path distribution. The right panels show transmittance and free-path distributions for increasing extinction values ($\times 10$, $\times 100$, $\times 1,000$, $\times 10,000$).

extinction by $\sigma'(\boldsymbol{x}) = a\sigma(\boldsymbol{x})$, where a is a positive constant. As a increases, $\sigma'(\boldsymbol{x})$ remains zero in the vacuum region, ensuring that the transmittance $T(\boldsymbol{u}, \boldsymbol{u} + \zeta\boldsymbol{\omega_u})$ stays at 1 as long as the path lies entirely in the vacuum. In contrast, $\sigma'(\boldsymbol{x})$ grows larger in the non-vacuum region, causing the transmittance to decay more rapidly.

As this process continues, $p_{\text{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta\boldsymbol{\omega_u})$ converges to the delta function $\delta(\boldsymbol{u} + \zeta\boldsymbol{\omega_u}, \boldsymbol{x}_s)$, where $\boldsymbol{x}_s$ denotes the first *limiting surface* (vacuum/non-vacuum interface) visible from the viewpoint.

This convergence can be visualized by considering a medium with a spherical distribution of extinction inside a dashed boundary (Figure 3). As the extinction increases, the transmittance diminishes more rapidly, and the peak of p_{fp} shifts toward the sphere's boundary. In the high extinction limit, p_{fp} becomes a sharp delta function at the first visible surface, effectively transitioning the medium to surface-like behavior. A formal mathematical proof is provided in supplementary material §14.

5 From Surface to Volume

To design scalar features and basis fields for volumetric media, we follow a key principle: as the extinction coefficient approaches infinity ($a \to \infty$), these features and fields should converge to their surface counterparts from the original stroke transfer framework. This is achieved by combining projection operators with the relationship between the delta function and the free-path distribution.

5.1 Projection and inverse-projection

Let Π and Π^{-1} denote the projection and inverse-projection operators, respectively. For a visible point $\boldsymbol{x}$ on the surface and its corresponding screen point $\boldsymbol{u}$, these operators establish the relationships $\boldsymbol{u} = \Pi(\boldsymbol{x})$ and $\boldsymbol{x} = \Pi^{-1}(\boldsymbol{u})$. If $\boldsymbol{x} = \Pi^{-1}(\boldsymbol{u})$, there exists a distance ζ such that $\boldsymbol{x} = \boldsymbol{u} + \zeta\boldsymbol{\omega_u}$, where $\boldsymbol{\omega_u}$ is the viewing direction at $\boldsymbol{u}$. The inverse-projection operator $\Pi^{-1}(\boldsymbol{u})$ can also be expressed as an integral along the line of sight:

$$\Pi^{-1}(\boldsymbol{u}) = \boldsymbol{u} + \zeta\boldsymbol{\omega_u} = \int_0^\infty \delta(\zeta, \zeta')(\boldsymbol{u} + \zeta'\boldsymbol{\omega_u})d\zeta'. \tag{4}$$

In the surface case, a screen feature $\phi_s(\boldsymbol{u})$ relates to a surface feature $\phi(\boldsymbol{x})$ via $\phi_s(\boldsymbol{u}) := \phi(\Pi^{-1}(\boldsymbol{u}))$, which can be reformulated as:

$$\phi_s(\boldsymbol{u}) = \int_0^\infty \delta(\zeta, \zeta')\phi(\boldsymbol{u} + \zeta'\boldsymbol{\omega_u})d\zeta'. \tag{5}$$

This expression weights the surface feature using the delta function δ, which isolates the contribution at the surface.

To generalize this formulation for volumetric media, we replace the delta function δ with the free-path distribution p_{fp}, resulting in:

$$\boxed{\phi_s(\boldsymbol{u}) := \int_0^\infty p_{\text{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta'\boldsymbol{\omega_u})\phi_v(\boldsymbol{u} + \zeta'\boldsymbol{\omega_u})d\zeta'.} \tag{6}$$

This accounts for volumetric features ϕ_v along the line of sight, weighted by their visibility as determined by p_{fp}.

In cases where the line of sight intersects both a medium and a surface, the relationship can be further generalized. Noting that (3) provides a natural weighting between volume and surface contributions, let $D(\boldsymbol{u})$ denote the distance to the first surface intersection along the ray. The following condition holds:

$$\int_0^{D(\boldsymbol{u})} p_{\text{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta'\boldsymbol{\omega_u})d\zeta' + T(\boldsymbol{u}, \boldsymbol{u} + D(\boldsymbol{u})\boldsymbol{\omega_u}) = 1. \tag{7}$$

Using this property, we extend the formulation to include contributions from both the volume and the surface:

$$\boxed{\begin{aligned}\phi_s(\boldsymbol{u}, \theta) &= \mathcal{I}(\boldsymbol{u}, \theta; p_{\text{fp}}, \phi_v)\\ &:= \int_0^{D(\boldsymbol{u})} p_{\text{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta'\boldsymbol{\omega_u})\phi_v(\boldsymbol{u} + \zeta'\boldsymbol{\omega_u}, \theta)d\zeta'\\ &\quad + T(\boldsymbol{u}, \boldsymbol{u} + D(\boldsymbol{u})\boldsymbol{\omega_u}))\phi_v(\boldsymbol{u} + D(\boldsymbol{u})\boldsymbol{\omega_u}, \theta).\end{aligned}} \tag{8}$$

Here, we introduced an auxiliary variable θ, which will find useful to accommodate angular dependencies for radiance computations in (9) and (10). $\mathcal{I}(\boldsymbol{u}, \theta; p_{\text{fp}}, \phi_v)$ provides a unified framework for handling features that span both the volume and the surface. In the subsequent sections, we detail how this formulation is applied to extend the computation of features and basis fields for participating media.

5.2 Features

5.2.1 Intensity and apparent intensity gradient. In the stroke transfer framework for surfaces, diffuse and specular components are treated separately, with only luminosity (excluding color information) considered. For volumetric media, we integrate diffuse and specular components, typically encoded using a single phase function, and include color information alongside luminosity.

For volumetric media, the raw screen intensity $\boldsymbol{I}_s(\boldsymbol{u})$ for each color channel c is computed as:

$$\boldsymbol{I}_s^{(c)}(\boldsymbol{u}) := \boldsymbol{L}_i^{(c)}(\boldsymbol{u}, \boldsymbol{\omega_u}) = \mathcal{I}(\boldsymbol{u}, -\boldsymbol{\omega_u}; p_{\text{fp}}^{(c)}, \boldsymbol{I}^{(c)}), \tag{9}$$

where the outgoing radiance $\boldsymbol{I}^{(c)}(\boldsymbol{x}, -\boldsymbol{\omega_u})$ is given by:

$$\begin{aligned}\boldsymbol{I}^{(c)}(\boldsymbol{x}, \boldsymbol{\omega}_o) &= (1 - \alpha^{(c)}(\boldsymbol{x}))\boldsymbol{L}_e^{(c)}(\boldsymbol{x}, \boldsymbol{\omega}_o)\\ &\quad + \alpha^{(c)}(\boldsymbol{x}) \int_{\mathcal{S}^2} \rho_{\boldsymbol{\omega}_o}^{(c)}(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)\boldsymbol{L}_i^{(c)}(\boldsymbol{x}, \boldsymbol{\omega}_i)d\boldsymbol{\omega}_i.\end{aligned} \tag{10}$$

Here, $\boldsymbol{L}_e^{(c)}(\boldsymbol{x}, \boldsymbol{\omega}_o)$ represents the self-emission term. Together, (9) and (10) constitute the rendering equation for participating media.

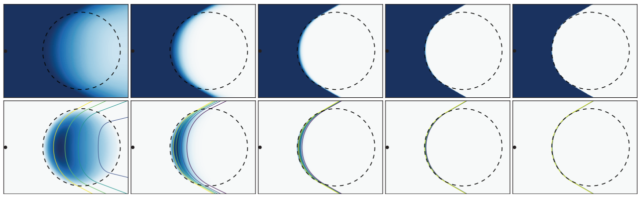


Fig. 4. **Convergence of transmittance and its level lines at high extinction limit**. For the extinction setting shown in Figure 3, we place the viewpoint at the black dot and compute the transmittance for points in the space (top). The corresponding isocontours of the transmittance and the free-path distribution are shown below (bottom). From left to right, the extinction is scaled by factors of 1, 10, 100, 1,000, and 10,000. As the extinction increases, the level lines of the transmittance converge to the sphere's surface, while the free-path distribution approaches a delta function at the surface. Consequently, the apparent curvature converges to the curvature of the sphere.

The primary objective of using the rendering equation is to distinguish various visual cues. While strict accuracy is not essential, consistency is prioritized. Biased methods like ray marching or approximations like single scattering suffice for our results.

The computed intensity is converted to the L*a*b* color space. We use the a* and b* components as well as the tone-mapped luminance $\tilde{I}_s(\boldsymbol{u})$ as intensity features. The tone mapping operator follows the approach in Todo et al. [2022]. For the apparent intensity gradient, we compute the screen-space gradient as $\tilde{I}_s^{\nabla_2}(\boldsymbol{u}) := |\nabla_2 \tilde{I}_s(\boldsymbol{u})|$.

5.2.2 Apparent Gaussian and mean curvatures. Gaussian and mean curvatures within the medium are computed from the isosurfaces of a scalar function $\psi(\boldsymbol{x})$ and integrated along the line of sight to yield screen-space curvatures. Selecting an appropriate $\psi(\boldsymbol{x})$ is key. While candidates include the extinction coefficient $\sigma(\boldsymbol{x})$ and negated[1] transmittance $-T(\Pi(\boldsymbol{x}), \boldsymbol{x})$, we adopt the latter due to its smoother behavior, particularly in thin media, as it is less sensitive to noise. Although transmittance may vary with wavelength, we use its average for simplicity.

From the discretely sampled $\psi(\boldsymbol{x})$, first- and second-order derivatives are computed using cubic splines [Stomakhin et al. 2013], and Goldman's formulae [2005] yield the Gaussian curvature $\kappa_G(\boldsymbol{x})$ and mean curvature $\kappa_m(\boldsymbol{x})$. Screen-space curvatures are computed as:

$$\kappa_{*s}(\boldsymbol{u}) := \mathcal{I}(\boldsymbol{u}; p_{\mathrm{fp}}, \kappa_*). \tag{11}$$

$\kappa_{*s}(\boldsymbol{u})$ naturally tend to zero in thin media without background surfaces, aligning with human perception (e.g., air density fluctuations do not create perceivable curvature).

5.2.3 Apparent normals. Using $\psi(\boldsymbol{x}) = -T(\Pi(\boldsymbol{x}), \boldsymbol{x})$, we compute the object-space normal as $\boldsymbol{n}(\boldsymbol{x}) = -\frac{\nabla\psi(\boldsymbol{x})}{|\nabla\psi(\boldsymbol{x})|}$. The screen-space normal $\boldsymbol{n}_s(\boldsymbol{u})$ is obtained using the model-view matrix $\mathbf{T}^{\mathrm{mv}}$ as:

$$\boldsymbol{n}_s(\boldsymbol{u}) := \mathbf{T}^{\mathrm{mv}-\top}\mathcal{I}(\boldsymbol{u}; p_{\mathrm{fp}}, \boldsymbol{n}). \tag{12}$$

We use the x-, y-, and z-components of $\boldsymbol{n}_s(\boldsymbol{u})$ as the scalar features for the apparent normal.

[1]Negation is applied because transmittance is a decreasing function.

5.2.4 Temperature. We compute the screen-space temperature as

$$C_s(\boldsymbol{u}) := \mathcal{I}(\boldsymbol{u}; p_{\mathrm{fp}}, C). \tag{13}$$

5.2.5 Apparent relative velocity. For simplicity, we assume a fixed grid in space to represent the time-varying flow of the medium. Specifically, the center position of the j-th grid cell, $\boldsymbol{p}_j$, is assumed to remain fixed over time, while the velocity at this cell in the n-th frame is denoted by $\boldsymbol{v}_j^{(n)}$. Let Δt represent the interval between consecutive frames. The relative velocity $\hat{\boldsymbol{v}}^{(n)}(\boldsymbol{p}_j)$ of each cell on the screen is then estimated as: $\hat{\boldsymbol{v}}^{(n)}(\boldsymbol{p}_j) := \frac{\Pi^{(n+1)}(\boldsymbol{p}_j + \Delta t \boldsymbol{v}_j^{(n)}) - \Pi^{(n)}(\boldsymbol{p}_j)}{\Delta t}$, where $\boldsymbol{p}_j + \Delta t \boldsymbol{v}_j^{(n)}$ is the predicted position of the cell due to advection, and $\Pi^{(n)}$ denotes the camera projection for the n-th frame, allowing us to account for camera motion. This per-cell relative velocity is interpolated in the object space to produce a continuous relative velocity field. The relative velocity at a surface point is computed similarly. The screen-space relative velocity feature is then computed as:

$$\hat{\boldsymbol{v}}_s^{(n)}(\boldsymbol{u}) := \mathcal{I}(\boldsymbol{u}; p_{\mathrm{fp}}, \hat{\boldsymbol{v}}^{(n)}). \tag{14}$$

We also include its norm $|\hat{\boldsymbol{v}}_s^{(n)}(\boldsymbol{u})|$ as a feature.

5.2.6 Transmittance. We have the transmittance feature $T_s(\boldsymbol{u})$ as

$$T_s(\boldsymbol{u}) := 1 - \exp\left(\int_0^\infty p_{\mathrm{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta'\boldsymbol{\omega}_{\boldsymbol{u}}) d\zeta'\right). \tag{15}$$

If the line of sight intersects a surface, the transmittance is set to 0.

5.2.7 Apparent mean free-path. In optically or geometrically thick media, objects behind the medium are completely occluded, resulting in near-zero transmittance in most inner regions. To enable the system to distinguish such regions, we define the apparent mean free-path feature, d_s, as the expected mean free-path ($\frac{1}{\sigma}$) along the line of sight until the ray intersects an object:

$$d_s(\boldsymbol{u}) := \int_0^{D(\boldsymbol{u})} p_{\mathrm{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta'\boldsymbol{\omega}_{\boldsymbol{u}}) \frac{1}{\sigma(\boldsymbol{u} + \zeta'\boldsymbol{\omega}_{\boldsymbol{u}}) + \sigma_\varepsilon} d\zeta', \tag{16}$$

where $\sigma_\varepsilon = 1/2000$ (chosen to approximate the extinction of air) is introduced to bound the apparent mean free-path: $0 \le d_s \le 2000$.

5.2.8 Distance from silhouettes. The distance from silhouettes is an exception to (6) or (5.1). For surfaces, we compute the signed distance function (SDF) $\xi_O(\boldsymbol{u})$, where $\xi_O(\boldsymbol{u}) = 0$ at silhouette lines, positive for the interior, and negative for the exterior. The absolute value gives the distance to silhouette lines. For media, a similar SDF $\xi_M(\boldsymbol{u})$ is defined, and the combined scalar feature is $\xi_s(\boldsymbol{u}) = \max(\xi_O(\boldsymbol{u}), \xi_M(\boldsymbol{u}))$.

One way to compute $\xi_M(\boldsymbol{u})$ is to first compute the transmittance $T_s^{(M)}(\boldsymbol{u}) := 1 - \exp\left(\int_0^{D(\boldsymbol{u})} p_{\mathrm{fp}}(\boldsymbol{u}, \boldsymbol{u} + \zeta'\boldsymbol{\omega}_{\boldsymbol{u}}) d\zeta'\right)$, for the visible part of the medium and set $\xi_M(\boldsymbol{u}) = 0$ at $T_s^{(M)}(\boldsymbol{u}) = \eta$ for a threshold η. However, using a single threshold can cause popping artifacts (Figure 5) when regions rapidly change opacity and the segmentation due to $T_s^{(M)}(\boldsymbol{u}) = \eta$ changes topology. To reduce sensitivity,

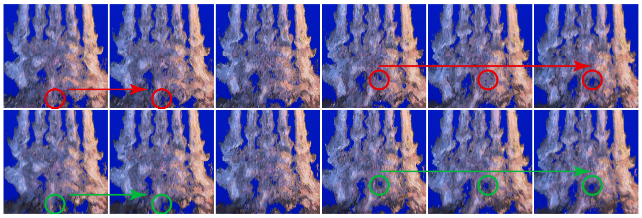


Fig. 5. **Comparison of distance from silhouettes computation**. Using a single threshold (top) causes abrupt changes in strokes due to segmentation artifacts. The integral approach (bottom) eliminates these artifacts.

we compute the expectation:

$$\xi_M(\boldsymbol{u}) = \int_0^1 \xi_M(\boldsymbol{u}, \eta) p_{\text{dist}}(\eta) d\eta, \tag{17}$$

where $p_{\text{dist}}(\eta)$ is a probability density over thresholds. Details on $p_{\text{dist}}(\eta)$ and efficient computation of $\xi_M(\boldsymbol{u})$ are in supplementary material §3. In the high-extinction limit, regions with $0 < T_s(\boldsymbol{u}) < 1$ converge to silhouette lines, including the surface-only case.

5.3 Standardization of features

We standardize features following Todo et al. [2022], with key updates: 1) correcting the scaling of Gaussian and mean curvatures to reflect their behavior under coordinate transformations and 2) adding standardization for temperature, transmittance, apparent mean free-path, and apparent relative velocity. Detailed formulae are in supplementary material §4.

5.4 Basis fields

Building on the six basis fields introduced by Todo et al. [2022]—intensity gradient and its 90° rotation, silhouette-guided direction and its 90° rotation, and apparent normal and its 90° rotation—we extend the framework by adding four new basis fields: apparent relative velocity and its 90° rotation, as well as the gradient of apparent mean free-path and its 90° rotation. These additions account for the non-rigid flow characteristics of participating media. Details on the computation of these ten basis fields in screen space are provided in the supplementary material §5.

6 Choosing Exemplar Frames

Features are computed for each animation frame of the 3D scene, requiring approximately 7 seconds per frame with Taichi [Hu et al. 2019] acceleration.

To minimize the number of exemplar frames while maintaining comprehensive feature coverage, we introduce an automatic selection algorithm based on Gaussian Mixture Models (GMMs) and Bayesian Information Criterion (BIC) [Schwarz 1978]. Each frame n is treated as a set of feature vectors $\boldsymbol{\Phi}^{(n)}$, where $\boldsymbol{\phi}_i^{(n)} \in \boldsymbol{\Phi}^{(n)}$ denotes the feature vector at pixel i. Let $\boldsymbol{\Phi}^{\text{all}}$ denote the union of feature samples across all frames. To reduce computational cost for exemplar selection, we downsample feature maps to 128×128 and sample every sixth frame.

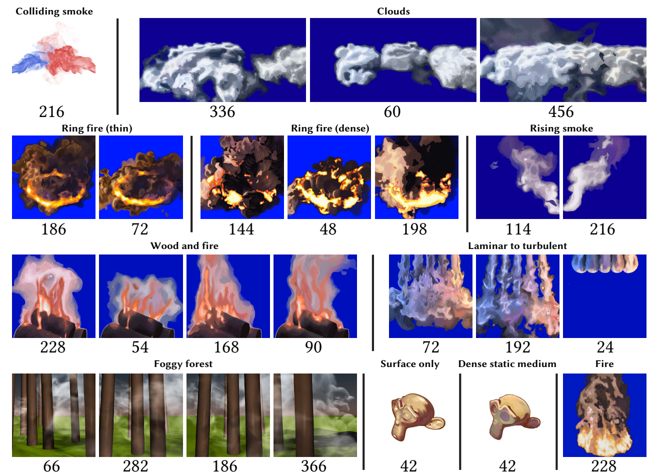


Fig. 6. **Chosen exemplars for each scene**. The numbers shown below each images are the frame indices of the exemplar frames.

For each frame, a GMM $p_{\text{feature}}^{(n)}(\boldsymbol{\phi}; \Theta^{(n)})$ is fitted to approximate the feature distribution:

$$p_{\text{feature}}^{(n)}(\boldsymbol{\phi}; \Theta^{(n)}) = \sum_{k=1}^{N_{\text{GMM}}} w_k^{(n)} \mathcal{G}(\boldsymbol{\mu}_k^{(n)}, \boldsymbol{\Sigma}_k^{(n)}), \tag{18}$$

where $\Theta^{(n)}$ consists of weights $\{w_k^{(n)}\}$, means $\{\boldsymbol{\mu}_k^{(n)}\}$, and covariances $\{\boldsymbol{\Sigma}_k^{(n)}\}$, with $\sum_k w_k^{(n)} = 1$, and $\mathcal{G}$ is the Gaussian kernel. We fix $N_{\text{GMM}} = 15$ for all frames.

The probability density for a set of exemplar frames $\mathcal{E}$ is defined as the average of their individual GMMs:

$$p_{\text{feature}}^{\mathcal{E}}(\boldsymbol{\phi}; \mathcal{E}, \{\Theta^{(n)}\}) := \frac{1}{|\mathcal{E}|} \sum_{n \in \mathcal{E}} p_{\text{feature}}^{(n)}(\boldsymbol{\phi}; \Theta^{(n)}). \tag{19}$$

The goal is to select exemplars that maximizes the likelihood

$$\mathcal{L}(p_{\text{feature}}^{\mathcal{E}}, \boldsymbol{\Phi}^{\text{all}}) := \log\left(\prod_{n=1}^{N_{\text{frames}}} \prod_{i=1}^{N_{\text{pts}}} p_{\text{feature}}^{\mathcal{E}}(\boldsymbol{\phi}_i^{(n)})\right), \tag{20}$$

while incorporating a penalty for model complexity. To balance model fit and complexity, we minimize the information criterion

$$E(\mathcal{E}) := \lambda N_{\text{params}}(\mathcal{E}) \log(|\boldsymbol{\Phi}^{\text{all}}|) - 2\mathcal{L}(p_{\text{feature}}^{\mathcal{E}}, \boldsymbol{\Phi}^{\text{all}}), \tag{21}$$

where the number of parameters $N_{\text{params}}(\mathcal{E})$ is given by

$$N_{\text{params}}(\mathcal{E}) = |\mathcal{E}| N_{\text{GMM}} \left(1 + d_{\boldsymbol{\phi}} + \frac{d_{\boldsymbol{\phi}}(d_{\boldsymbol{\phi}} + 1)}{2}\right). \tag{22}$$

The regularization constant $\lambda > 0$ adjusts for the singularity of GMMs [Watanabe 2013]. We set $\lambda = \lambda_{\text{M}}/35$, where $\lambda_{\text{M}} = \frac{1}{2}(N_{\text{GMM}} - 1)d_{\boldsymbol{\phi}}(d_{\boldsymbol{\phi}} + 3)$ is the theoretical upper bound. Our experiments show the method is robust to the choice of λ.

Exhaustive search over all frame subsets is computationally intractable. Therefore, we adopt (1) *early stopping* when $E(|\mathcal{E}| + 1) > E(|\mathcal{E}|)$ and (2) *greedy selection*, which fixes previously chosen frames and searches for one additional frame at each step, reducing the complexity to $O(N_{\text{frames}}|\mathcal{E}|)$.

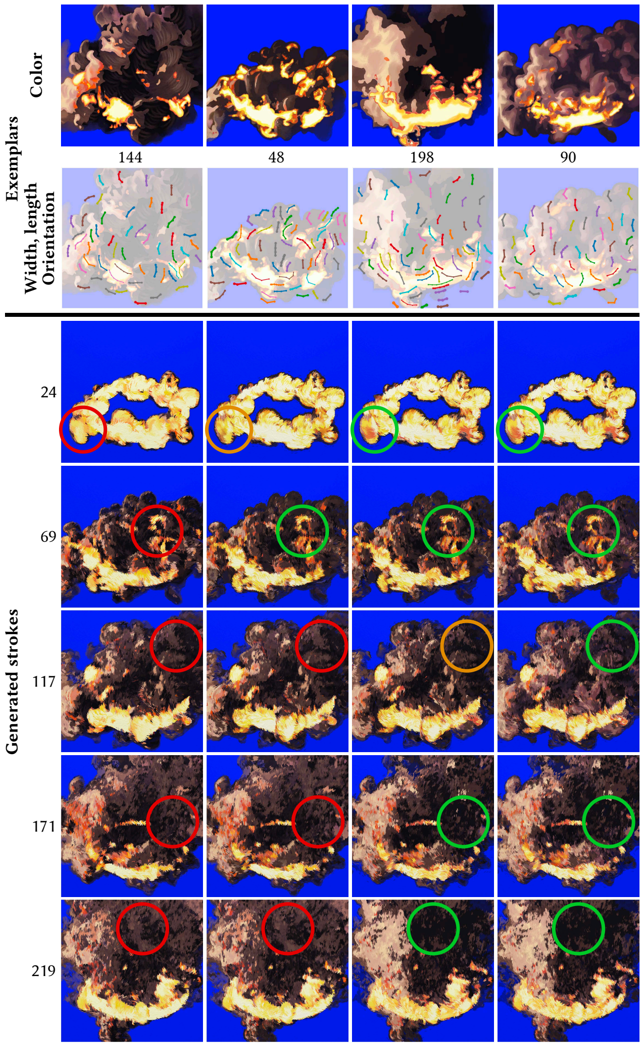


Fig. 7. **Impact of exemplar frame count on generated results**. At the top, we display exemplars, each consisting of a color image along with annotated widths, colors, and orientations, for the 144th, 48th, 198th, and 90th frames. These frames correspond to the first, second, third, and fourth frames automatically selected by our approach, with the fourth frame specifically chosen by using a lower λ parameter. At the bottom, we show the generated strokes for the 24th, 69th, 117th, 171st, and 219th frames using different combinations of exemplars: only the first (frame 144, leftmost), the first and second (frames 144 and 48, middle left), the first three (frames 144, 48, and 198, middle right), and all four exemplars (frames 144, 48, 198, and 90, rightmost). Significant improvements are observed when adding the second and third exemplars, with diminishing returns from including the fourth.

The number of selected exemplars vary from 1 to 4 per scene (Figure 6). Our method effectively captures diverse scenarios, such as different smoke locations in Rising-Smoke or different density stages in Ring-Fire (Dense). Figure 7 shows the impact of the number of exemplars. Reducing λ by a factor of 10 results in four exemplars for all scenes, further demonstrating the robustness of the method.

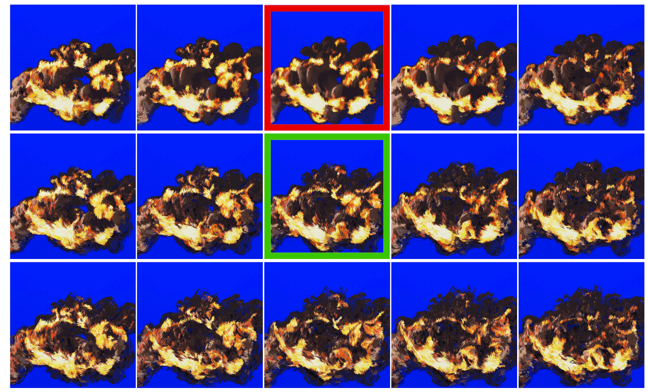


Fig. 8. **Effect of displacement for feature query**. In a sequence of generated animation frames exhibiting overfitting (top, with reduced fluctuations in the middle frames), the displacement approach (with $r^{\text{min}}_{\text{scale}} = 0$ and $r^{\text{o-f}}_{\text{scale}} = 0.04$) effectively suppresses this effect (middle). Further increasing $r^{\text{min}}_{\text{scale}}$ introduces added randomness to the animation (bottom).

7 Exemplar Collection and Regression

For the selected set of exemplar frames, $\mathcal{E}$, the user is asked to paint the exemplars. We encourage the user to suppress randomness inherent to discrete stroke representations. While users can freely choose colors, widths, lengths, and orientations, we recommend that these elements be consistently aligned with the features. This consistency serves as a reliable baseline, making it easier to introduce or control randomness later if needed. Conversely, attempting to recover this baseline by reversing randomness is inherently an ill-posed problem. An added benefit is that our system becomes accessible to users with varying skill levels, even those without expertise in techniques like color divisionism.

Using the painted exemplar frames, along with the computed features and basis fields for the given 3D scene, we train the models $\mathbf{M}_{\text{c,w,l}}$ and $\mathbf{M}_{\text{v}}$. The training process takes approximately 3 minutes. The learned models are then applied to transfer colors, widths, lengths, and orientations to all frames in the animation, with the transfer process requiring approximately 3 seconds per frame. The transferred four attributes vary consistently over frames, as shown in our supplementary video.

8 Controlling Fluctuations

To control randomness, we apply displacement during feature queries (Figure 8) by shifting the query from a pixel at $\boldsymbol{u}$ to a displaced position $\boldsymbol{u}' = \boldsymbol{r}(\boldsymbol{u})$. The displacement map $\boldsymbol{r}$, modeled with octave (fractal) noise [Perlin 1985] for spatial coherence, is defined as $\boldsymbol{r} = r_{\text{scale}}\hat{\boldsymbol{r}}$, where $\hat{\boldsymbol{r}}$ (direction) and r_{scale} (magnitude) use two octave noises for $\hat{\boldsymbol{r}}$ and one for r_{scale}.

Overfitting near exemplar frames can arise (when using nearest neighbor regression for colors) due to minor user style inconsistencies, which are amplified in frames farther from the exemplars. To mitigate this, we scale r_{scale} based on frame distance $\Delta^{(j)}$ to the nearest exemplar frame:

$$r^{(j)}_{\text{scale}} = r^{\text{min}}_{\text{scale}} + r^{\text{o-f}}_{\text{scale}} \exp(-r_{\Sigma}\Delta^{(j)}). \tag{23}$$

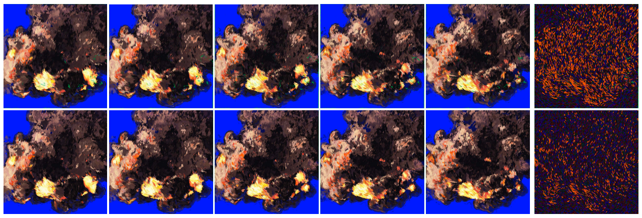


Fig. 9. **Comparison of generated strokes**. We compare strokes generated by the method of Todo et al.[2022] (top) with those produced by our approach (bottom). Void regions—where stroke generation failed to fill—are highlighted in green. Newly inserted strokes are shown in orange in the rightmost images. The proportion of affected pixels is 17.4% for Todo et al.[2022] and 6.5% for our method.

where $r^{\text{o-f}}_{\text{scale}} = 0.04$ and $r_{\Sigma} = 8.0$ (if the corresponding attribute is learned and transferred via nearest neighbor regression, and $r^{\text{o-f}}_{\text{scale}} = 0.0$ if linear regression is used). Adding a constant $r^{\text{min}}_{\text{scale}}$ further introduces controlled randomness.

For the orientation field, displacement is applied during feature queries, but not for the basis fields themselves. Coefficients derived from the learned map are used to combine basis fields into an unsmoothed orientation field, which is then smoothed spatially and temporally (supplementary material §6). Randomness in stroke directions is introduced as an angular offset relative to the smoothed orientation field.

9 Stroke Rendering

Our stroke generation algorithm uses anchor points to maintain temporal coherence, storing random values (e.g., angle offsets) and the four stroke attributes—color, width, length, and orientation—sampled from their respective fields to generate strokes. Unlike Todo et al. [2022], which employs a hierarchical structure for anchor points, our method places anchor points and generates strokes directly on the visible screen region, coupling these processes. Anchor points from the previous frame are advected along the relative velocity field using the TVD-RK3 scheme [Gottlieb and Shu 1998], simulating stroke motion. New anchor points are sampled in void regions, with strokes generated immediately. Void regions are updated iteratively until no additional strokes can be inserted. Invisible strokes are deleted, but those likely to reappear are retained by applying a slightly shrunk mask during overlap detection, minimizing popping artifacts. This strategy produces denser strokes than Todo et al.[2022], reducing reliance on an undercoat background layer to cover void regions and improving temporal coherence by inserting fewer strokes per frame (Figure9). The computation time roughly depends on the number of strokes generated. It takes approximately 30 seconds per frame for 80, 000 strokes.

For transparent styles with plausible (Kubelka-Munk-based) color blending, a post-processing step optimizes stroke colors for realistic mixing [Sochorová and Jamriška 2021], ensuring that the mixed colors align with the transferred colors, enhancing visual consistency (Figure 10).

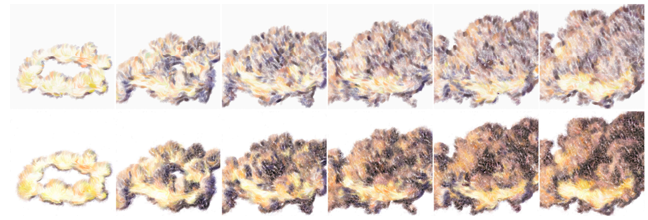
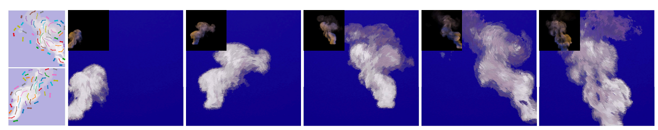


Fig. 10. **Effectiveness of color optimization for plausible color blending**. Top: Results without color optimization. Bottom: Results with color optimization, showing improved and consistent blending effects.

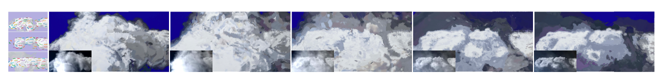
10 Results

We demonstrate the versatility of our method across a variety of scenarios. All results were generated on a laptop equipped with an Apple Silicon M4 Max CPU. Code and data are available at the `Stroke_Transfer_For_Participating_Media` repository. Scene statistics are provided in supplementary material §8, timings in §9, and animations in our supplementary video.

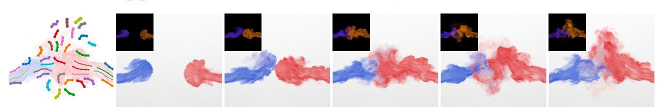
Rising Smoke. A single smoke rises under animated lighting and camera motion. Strokes capture the swirling flow and smoothly vary across dense and sparse regions, maintaining stylistic consistency. Exemplars (and annotations) are shown in the left. The inset images show (tone-mapped) intensity features (rendered results).



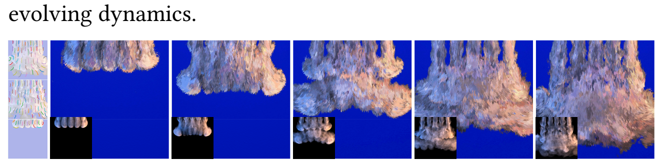
Clouds. A swarm of clouds flows toward the viewer, with dense clusters rising and thinning as they dissipate. The use of opaque strokes vividly conveys the evolving cloud structure.



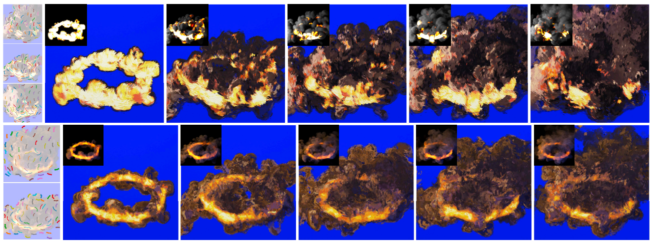
Colliding Smokes. Multiple colored smokes collide. By leveraging additional features from a* and b* channels, our method distinguishes between materials and captures natural mixing post-collision. Exemplars were taken directly from rendered frames with a color modulator applied, rather than drawn by a user.



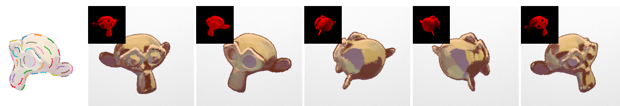
Laminar to Turbulent. The flow transitions from laminar to turbulent motion. Our method adapts seamlessly, faithfully capturing the evolving dynamics.



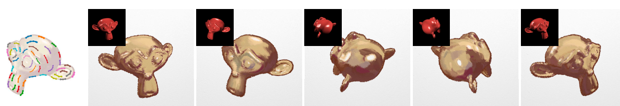
Ring Fire with dense (top) and thin (bottom) Settings. We show fire animations under different extinction coefficients. As the fire cools outward, black-body radiation weakens and dark smoke emerges. Our method captures this fire-to-smoke transition naturally.



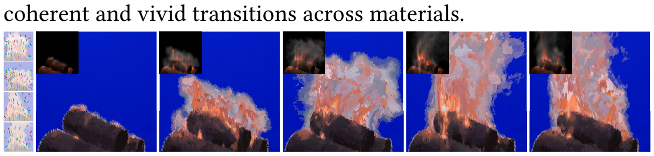
Dense Static Medium. A dense, static volumetric medium exhibits solid-like subsurface scattering. The style from the exemplar consistently transfers to the animated sequence.



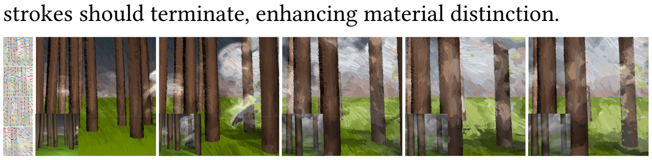
Surface Only. In scenes with only surface objects, our method successfully transfers style, despite minor temporal artifacts near boundaries due to screen-space stroke placement.



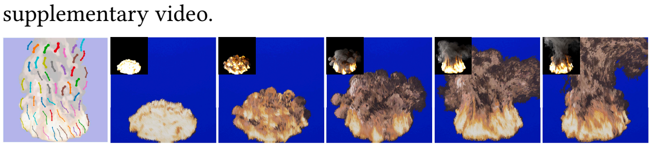
Wood and Fire. A hybrid scene with wood surfaces and fire. Our method distinguishes between wood, smoke, and flames, producing coherent and vivid transitions across materials.



Foggy Forest. Fog flows through a forest of trees and grass. Region labels (Figure 11), transferred similarly to colors, inform where strokes should terminate, enhancing material distinction.



Fire. An alternative fire setup where strong vertical flow stretches fire and smoke upward. We also demonstrate different stroke shaders mimicking plausible pencil and charcoal effects in Figure 1 and our supplementary video.



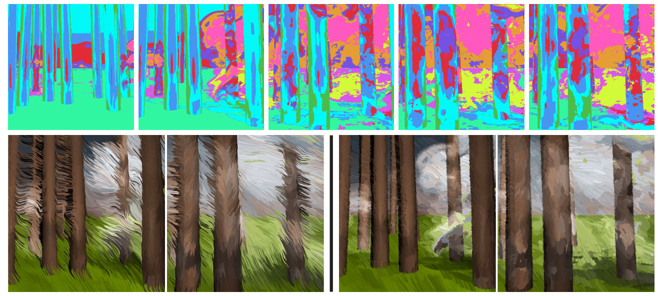


Fig. 11. **Effectiveness of region labels.** Top: transferred region labels. Bottom left and right: strokes generated without and with region labels, respectively.

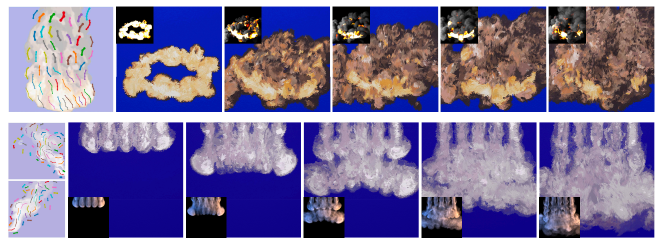


Fig. 12. **Cross scene transfer**. Top: from the exemplars for the Fire scene to the Ring-Fire (Dense) scene. Bottom: from the exemplars for the Rising-Smoke scene to the Laminar-to-Turbulent scene.

Cross-Scene Transfer. Figure 12 shows style transfer across different scenes. The results remain coherent and visually consistent.

Comparison to Previous Work. We compare against patch-based [Fišer et al. 2016; Texler et al. 2020], Neural Transfer [Ghiasi et al. 2017], stroke-based neural methods [Hu et al. 2023; Kotovenko et al. 2021], and a diffusion-based text-to-video approach [Liu et al. 2024]. Patch-based methods suffer from stitching artifacts; neural approaches often fail to capture authentic brushstrokes and suffer from temporal instability. Due to limited public implementations, conditional guidance (e.g., depth inputs) was not available for the text-to-video comparison. For details, please refer to our supplementary material §10 and supplementary video.

Additional Studies. Supplementary material §11 shows the learned matrices $\mathbf{M}_v$ and scene-specific feature importance. Ablation studies (Figure 13 and our supplementary video) show that removing illumination features causes noisier attributes, while removing normals, curvatures, or volumetric cues makes colors more uniform. Removing velocity features disrupts flow-aligned orientations, and removing silhouette features causes abrupt attribute changes. These results highlight the importance of adaptive feature selection. For the details of the ablation study settings, please see supplementary material §12. In supplementary material §13, we further validate attribute transfer accuracy by comparing transferred frames against additional exemplars. In the supplementary video, we also show variations under different stroke-length and width annotations.

Fig. 13. **Ablation study.** We compare our full model against versions with individual feature groups removed. Without illumination-related features, the generated colors become noisier; without normals, curvatures, or additional volumetric features, the results appear more monotonous; without the silhouette feature, artifacts emerge (highlighted by the red circle); and without velocity-related features, stroke orientations in vortex regions (red circle) fail to align properly with the flow.

11 Conclusions, Limitations, and Future Work

We have introduced a method for generating stroke-based painterly drawings for participating media, such as smoke, fire, and clouds, by transferring key stroke attributes—color, width, length, and orientation—from exemplar frames to animation frames. By extending feature and basis field computations to volumetric scenes, our approach naturally incorporates surface-based cases as a special instance. The results showcase versatility across thin and thick media, scenes with surfaces, self-emitting phenomena, and dynamic flows ranging from laminar to turbulent. Additionally, our formulation for geometric features, such as apparent curvatures, may inspire further advancements in patch-based or neural-based methods for participating media.

Our work has limitations. Restricting to four stroke attributes leaves other artistic factors unexplored, and expanding the attribute set could enable more nuanced effects. While our randomness modeling mitigates overfitting and introduces variation, it lacks fine-grained control. Learning and modeling randomness from exemplars, including stroke color correlations, could enhance fidelity and flexibility. Our single layer assumption uses a single layer of attributes for stroke placement. It would be interesting to explore the use of multiple layers of attributes for more expressiveness (e.g., allowing strokes to pass through each other) and reduced "fluffy" effects (e.g., near apparent surface-volume interface). Extending the method to handle non-exponential media and incorporating full light transport with spectral effects could improve realism. Future work could also explore shape simplification (e.g., sumi-e or cubism) and other artistic styles, such as hatching.

Acknowledgments

We thank the anonymous reviewers for their insightful suggestions and discussions. We thank Mitsuki Hamamichi and Yuta Morimoto for helping us preparing the examples. This work was supported in part by a grant from JST FOREST Program, JPMJFR206R, Japan, a JSPS Grant-in-Aid for Scientific Research (S) 25H00399, Japan, a JSPS Grant-in-Aid for Scientific Research (B) 25K00921, Japan, a JSPS Grant-in-Aid for Scientific Research (C) 22K12051, and a JSPS Grant-in-Aid for Scientific Research (C) 25K15140.

References

Eduardo J. Álvarez, Celso Campos, Silvana G. Meire, Ricardo Quirós, Joaquin Huerta, and Michael Gould. 2007. Interactive Cartoon Rendering and Sketching of Clouds and Smoke. In *Computational Science – ICCS 2007*, Yong Shi, Geert Dick van Albada, Jack Dongarra, and Peter M. A. Sloot (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 138–145.

Connelly Barnes and Fang-Lue Zhang. 2017. A survey of the state-of-the-art in patch-based synthesis. *Computational Visual Media* 3, 1 (2017), 3–20. doi:10.1007/s41095-016-0064-2

Bill Baxter, Vincent Scheib, Ming C. Lin, and Dinesh Manocha. 2001. DAB: Interactive Haptic Painting with 3D Virtual Brushes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 461—-468. doi:10.1145/383259.383313

Beer. 1852. Bestimmung der Absorption des rothen Lichts in farbigen Flüssigkeiten. *Annalen der Physik* 162, 5 (1852), 78–88. doi:10.1002/andp.18521620505 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.18521620505

Pierre Bénard, Forrester Cole, Michael Kass, Igor Mordatch, James Hegarty, Martin Sebastian Senn, Kurt Fleischer, Davide Pesare, and Katherine Breeden. 2013. Stylizing Animation by Example. *ACM Transactions on Graphics* 32, 4 (Proc. of SIGGRAPH 2013), Article 119 (jul 2013), 12 pages. doi:10.1145/2461912.2461929

Benedikt Bitterli, Srinath Ravichandran, Thomas Müller, Magnus Wrenninge, Jan Novák, Steve Marschner, and Wojciech Jarosz. 2018. A radiative transfer framework for non-exponential media. *ACM Trans. Graph.* 37, 6, Article 225 (dec 2018), 17 pages. doi:10.1145/3272127.3275103

Pierre Bouguer. 1729. *Essai d'optique sur la gradation de la lumière.* Claude Jombert, 16–22.

Robert Bridson. 2008. *Fluid Simulation.* A. K. Peters, Ltd., USA.

Mark Browning, Connelly Barnes, Samantha Ritter, and Adam Finkelstein. 2014. Stylized keyframe animation of fluid simulations. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering* (Vancouver, British Columbia, Canada) *(NPAR '14)*. Association for Computing Machinery, New York, NY, USA, 63–70. doi:10.1145/2630397.2630406

Luis Cardona and Suguru Saito. 2015. Hybrid-Space Localized Stylization Method for View-Dependent Lines Extracted from 3D Models. In *Proceedings of the Workshop on Non-Photorealistic Animation and Rendering (NPAR '15)*. Eurographics Association, Goslar, DEU, 79–89.

Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J. Seron, and François X. Sillion. 2005. A survey on participating media rendering techniques. *The Visual Computer* 21, 5 (2005), 303–328. doi:10.1007/s00371-005-0287-1

Subrahmanyan Chandrasekhar. 1950. *Radiative transfer.* Oxford University Press.

Hila Chefer, Shiran Zada, Roni Paiss, Ariel Ephrat, Omer Tov, Michael Rubinstein, Lior Wolf, Tali Dekel, Tomer Michaeli, and Inbar Mosseri. 2024. Still-Moving: Customized Video Generation without Customized Video Data. *ACM Trans. Graph.* 43, 6, Article 244 (Nov. 2024), 11 pages. doi:10.1145/3687945

Zhili Chen, Byungmoon Kim, Daichi Ito, and Huamin Wang. 2015. Wetbrush: GPU-Based 3D Painting Simulation at the Bristle Level. *ACM Transactions on Graphics* 34, 6 (Proc. of SIGGRAPH ASIA 2015), Article 200 (oct 2015), 11 pages. doi:10.1145/2816795.2818066

Frédo Durand. 2002. An Invitation to Discuss Computer Depiction. In *Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '02)*. Association for Computing Machinery, New York, NY, USA, 111—-124. doi:10.1145/508530.508550

Ashley M. Eden, Adam W. Bargteil, Tolga G. Goktekin, Sarah Beth Eisinger, and James F. O'Brien. 2007. A method for cartoon-style rendering of liquid animations. In *Proceedings of Graphics Interface 2007* (Montreal, Canada) *(GI '07)*. Association for Computing Machinery, New York, NY, USA, 51–55. doi:10.1145/1268517.1268528

Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sýkora. 2016. StyLit: Illumination-Guided Example-Based Stylization of 3D Renderings. *ACM Transactions on Graphics* 35, 4 (Proc. of SIGGRAPH 2016), Article 92 (jul 2016), 11 pages. doi:10.1145/2897824.2925948

David Futschik, Michal Kučera, Michal Lukáč, Zhaowen Wang, Eli Shechtman, and Daniel Sýkora. 2021. STALP: Style Transfer with Auxiliary Limited Pairing. *Computer Graphics Forum* 40, 2 (Proc. of EUROGRAPHICS 2021) (2021), 563–573.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2414–2423. doi:10.1109/CVPR.2016.265

Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. arXiv:1705.06830 [cs.CV] https://arxiv.org/abs/1705.06830

Ron Goldman. 2005. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. Des.* 22, 7 (oct 2005), 632–658.

Sigal Gottlieb and Chi-Wang Shu. 1998. Total variation diminishing Runge-Kutta schemes. *Math. Comput.* 67, 221 (jan 1998), 73–85. doi:10.1090/S0025-5718-98-00913-2

Jie Guo, Mengtian Li, Zijing Zong, Yuntao Liu, Jingwu He, Yanwen Guo, and Ling-Qi Yan. 2021. Volumetric appearance stylization with stylizing kernel prediction network. *ACM Trans. Graph.* 40, 4, Article 162 (July 2021), 15 pages. doi:10.1145/3450626.3459799

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2024. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. In *The Twelfth International Conference on Learning Representations.* https://openreview.net/forum?id=Fx2SbBgcte

Paul Haeberli. 1990. Paint by numbers: abstract image representations. *Comput. Graph. (Proc. SIGGRPAH '90)* 24, 4 (Sept. 1990), 207–214. doi:10.1145/97880.97902

T. Haga, Henry Johan, and Tomoyuki Nishita. 2001. Animation Method for Pen-and-Ink Illustrations Using Stroke Coherency. In *Proc. of CAD & Graphics 2001.* 333–343.

James Hays and Irfan Essa. 2004. Image and video based painterly animation. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '04)*. Association for Computing Machinery, New York, NY, USA, 113–120. doi:10.1145/987657.987676

Siddharth Hegde, Christos Gatzidis, and Feng Tian. 2013. Painterly Rendering Techniques: A State-of-The-Art Review of Current Approaches. *Computer Animation and Virtual Worlds* 24, 1 (2013), 43–64. doi:10.1002/cav.1435

Aaron Hertzmann. 2003. Tutorial: A Survey of Stroke-Based Rendering. *IEEE Computer Graphics and Applications* 23, 4 (jul 2003), 70––81. doi:10.1109/MCG.2003.1210867

Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image Analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01).* Association for Computing Machinery, New York, NY, USA, 327––340. doi:10.1145/383259.383295

Aaron Hertzmann and Ken Perlin. 2000. Painterly rendering for video and interaction. In *Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '00)*. Association for Computing Machinery, New York, NY, USA, 7–12. doi:10.1145/340916.340917

Teng Hu, Ran Yi, Haokun Zhu, Liang Liu, Jinlong Peng, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. 2023. Stroke-based Neural Painting and Stylization with Dynamically Predicted Painting Region. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 7470–7480. doi:10.1145/3581783.3611766

Yuanming Hu, Tzu-Mao Li, Luke Anderson, Jonathan Ragan-Kelley, and Frédo Durand. 2019. Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Trans. Graph.* 38, 6, Article 201 (Nov. 2019), 16 pages. doi:10.1145/3355089.3356506

Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2015. LazyFluids: Appearance Transfer for Fluid Animations. *ACM Trans. Graph.* 34, 4, Article 92 (jul 2015), 10 pages. doi:10.1145/2766983

Ondřej Jamriška, Šárka Sochorová, Ondřej Texler, Michal Lukáč, Jakub Fišer, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Stylizing Video by Example. *ACM Transactions on Graphics* 38, 4 (Proc. of SIGGRAPH 2019), Article 107 (aug 2019), 11 pages. doi:10.1145/3306346.3323006

Adrian Jarabo, Carlos Aliaga, and Diego Gutierrez. 2018. A radiative transfer framework for spatially-correlated materials. *ACM Trans. Graph.* 37, 4, Article 83 (jul 2018), 13 pages. doi:10.1145/3197517.3201282

Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. 2003. Coherent Stylized Silhouettes. *ACM Transactions on Graphics* 22, 3 (Proc. of SIGGRAPH 2003) (jul 2003), 856––861. doi:10.1145/882262.882355

Evangelos Kalogerakis, Derek Nowrouzezahrai, Simon Breslav, and Aaron Hertzmann. 2012. Learning Hatching for Pen-and-Ink Illustration of Surfaces. *ACM Trans. Graph.* 31, 1, Article 1 (jan 2012), 17 pages. doi:10.1145/2077341.2077342

George Katanics and Tasso Lappa. 2003. Deep Canvas: Integrating 3D Painting and Painterly Rendering. In *Theory and Practice of Non-Photorealistic Graphics: Algorithms, Methods, and Production Systems (ACM SIGGRAPH 2003 Course Notes).*

Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. 2019. Transport-based neural style transfer for smoke simulations. *ACM Transactions on Graphics* 38, 6 (Nov. 2019), 1–11. doi:10.1145/3355089.3356560

Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. 2020. Lagrangian neural style transfer for fluids. *ACM Trans. Graph.* 39, 4, Article 52 (aug 2020), 10 pages. doi:10.1145/3386569.3392473

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Josh Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. 2024. VideoPoet: a large language model for zero-shot video generation. In *Proceedings of the 41st International Conference on Machine Learning* (Vienna, Austria) *(ICML'24)*. JMLR.org, Article 1005, 20 pages.

Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. 2022. A Survey on SPH Methods in Computer Graphics. *Computer Graphics Forum* 41, 2 (2022), 737–760. doi:10.1111/cgf.14508 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14508

Dmytro Kotovenko, Matthias Wright, Arthur Heimbrecht, and Bjorn Ommer. 2021. Rethinking Style Transfer: From Pixels to Parameterized Brushstrokes . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE Computer Society, Los Alamitos, CA, USA, 12191–12200. doi:10.1109/CVPR46437.2021.01202

Michael A. Kowalski, Lee Markosian, J. D. Northrup, Lubomir Bourdev, Ronen Barzel, Loring S. Holden, and John F. Hughes. 1999. Art-Based Rendering of Fur, Grass, and Trees. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99).* ACM Press/Addison-Wesley Publishing Co., USA, 433––438. doi:10.1145/311535.311607

Johann Heinrich Lambert. 1760. *Photometria sive de mensura et gradibus luminis, colorum et umbrae.* Eberhardt Klett.

Hochang Lee, Sanghyun Seo, Seungtaek Ryoo, and Kyunghyun Yoon. 2010. Directional texture transfer. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '10)*. Association for Computing Machinery, New York, NY, USA, 43–48. doi:10.1145/1809939.1809945

Gongye Liu, Menghan Xia, Yong Zhang, Haoxin Chen, Jinbo Xing, Yibo Wang, Xintao Wang, Ying Shan, and Yujiu Yang. 2024. StyleCrafter: Taming Artistic Video Diffusion with Reference-Augmented Adapter Learning. *ACM Trans. Graph.* 43, 6, Article 251 (Nov. 2024), 10 pages. doi:10.1145/3687975

Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. 2021. Paint Transformer: Feed Forward Neural Painting with Stroke Prediction . In *2021 IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE Computer Society, Los Alamitos, CA, USA, 6578–6587. doi:10.1109/ICCV48922.2021.00653

Xiao-Chang Liu, Yu-Chen Wu, and Peter Hall. 2023. Painterly Style Transfer With Learned Brush Strokes. *IEEE Transactions on Visualization and Computer Graphics* (2023), 1–12. doi:10.1109/TVCG.2023.3332950

Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Loring S. Holden, J. D. Northrup, and John F. Hughes. 2000. Art-Based Rendering with Continuous Levels of Detail. In *Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering (NPAR '00).* Association for Computing Machinery, New York, NY, USA, 59—-66. doi:10.1145/340916.340924

Morgan McGuire and Andi Fein. 2006. Real-time rendering of cartoon smoke and clouds. In *Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '06)*. Association for Computing Machinery, New York, NY, USA, 21–26. doi:10.1145/1124728.1124733

Barbara J. Meier. 1996. Painterly Rendering for Animation. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96).* Association for Computing Machinery, New York, NY, USA, 477—-484. doi:10.1145/237170.237288

Bailey Miller, Iliyan Georgiev, and Wojciech Jarosz. 2019. A null-scattering path integral formulation of light transport. *ACM Trans. Graph.* 38, 4, Article 44 (jul 2019), 13 pages. doi:10.1145/3306346.3323025

Florian Nolte, Andrew Melnik, and Helge Ritter. 2022. Stroke-based Rendering: From Heuristics to Deep Learning. arXiv:2302.00595 [cs.CV] https://arxiv.org/abs/2302.00595

J. D. Northrup and Lee Markosian. 2000. Artistic Silhouettes: A Hybrid Approach. In *Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering (NPAR '00).* Association for Computing Machinery, New York, NY, USA, 31—-37. doi:10.1145/340916.340920

Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo Methods for Volumetric Light Transport Simulation. *Computer Graphics Forum* 37, 2 (2018), 551–576. doi:10.1111/cgf.13383 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13383

Derek Nowrouzezahrai, Jared Johnson, Andrew Selle, Dylan Lacewell, Michael Kaschalk, and Wojciech Jarosz. 2011. A programmable system for artistic volumetric lighting. *ACM Trans. Graph.* 30, 4, Article 29 (2011), 8 pages. doi:10.1145/1964921.1964924

Sven C. Olsen, Bruce A. Maxwell, and Bruce Gooch. 2005. Interactive vector fields for painterly rendering. In *Proceedings of Graphics Interface 2005* (Victoria, British Columbia) *(GI '05)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 241–247.

Mark Pauly, Thomas Kollig, and Alexander Keller. 2000. Metropolis Light Transport for Participating Media. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000.* Springer-Verlag, Berlin, Heidelberg, 11–22.

Ken Perlin. 1985. An image synthesizer. *SIGGRAPH Comput. Graph.* 19, 3 (July 1985), 287–296. doi:10.1145/325165.325247

Max Planck. 1901. Ueber das Gesetz der Energieverteilung im Normalspectrum. *Annalen der Physik* 309, 3 (1901), 553–563. doi:10.1002/andp.19013090310 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.19013090310

Adam Platkevič, Cassidy Curtis, and Daniel Sýkora. 2021. Fluidymation: Stylizing Animations Using Natural Dynamics of Artistic Media. *Computer Graphics Forum* 40, 7 (2021), 21–32. doi:10.1111/cgf.14398 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14398

Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner. 2011. OverCoat: An Implicit Canvas for 3D Painting. *ACM Transactions on Graphics* 30, 4 (Proc. of SIGGRAPH 2011), Article 28 (jul 2011), 10 pages. doi:10.1145/2010324.1964923

Gideon Schwarz. 1978. Estimating the Dimension of a Model. *The Annals of Statistics* 6, 2 (1978), 461–464. http://www.jstor.org/stable/2958889

Andrew Selle, Alex Mohr, and Stephen Chenney. 2004. Cartoon rendering of smoke animations. In *Proceedings of the 3rd International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '04)*. Association for Computing Machinery, New York, NY, USA, 57–60. doi:10.1145/987657.987666

Mayank Singh and Scott Schaefer. 2010. Suggestive Hatching. In *Proceedings of the Sixth International Conference on Computational Aesthetics in Graphics, Visualization and Imaging* (London, United Kingdom) *(Computational Aesthetics'10)*. Eurographics Association, Goslar, DEU, 25—-32.

Šárka Sochorová and Ondřej Jamriška. 2021. Practical pigment mixing for digital painting. *ACM Trans. Graph.* 40, 6, Article 234 (Dec. 2021), 11 pages. doi:10.1145/3478513.3480549

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. Graph. (Proc. of SIGGRAPH 2013)* 32, 4 (2013), 102:1–10. doi:10.1145/2461912.2461948

Onkar Susladkar, Gayatri Deshmukh, Sparsh Mittal, and Parth Shastri. 2024. D2Styler: Advancing Arbitrary Style Transfer with Discrete Diffusion Methods. In *Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, December 1–5, 2024, Proceedings, Part VI* (Kolkata, India). Springer-Verlag, Berlin, Heidelberg, 63–82. doi:10.1007/978-3-031-78172-8_5

Daniel Sýkora, Ondřej Jamriška, Ondřej Texler, Jakub Fišer, Michal Lukáč, Jingwan Lu, and Eli Shechtman. 2019. StyleBlit: Fast Example-Based Stylization with Local Guidance. *Computer Graphics Forum* 38, 2 (Proc. of EUGRAPHICS 2019) (2019), 83–91.

Ondřej Texler, David Futschik, Michal kučera, Ondřej jamriška, Šárka Sochorová, Menclei Chai, Sergey Tulyakov, and Daniel SÝkora. 2020. Interactive Video Stylization Using Few-Shot Patch-Based Training. *ACM Transactions on Graphics* 39, 4 (Proc. of SIGGRAPH 2020), Article 73 (jul 2020), 11 pages. doi:10.1145/3386569.3392453

Hideki Todo, Kunihiko Kobayashi, Jin Katsuragi, Haruna Shimotahira, Shizuo Kaji, and Yonghao Yue. 2022. Stroke Transfer: Example-based Synthesis of Animatable Stroke Styles. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 54, 10 pages. doi:10.1145/3528233.3530703

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. VideoComposer: compositional video synthesis with motion controllability. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 334, 18 pages.

Sumio Watanabe. 2013. A widely applicable Bayesian information criterion. *J. Mach. Learn. Res.* 14, 1 (March 2013), 867–897.

Chung-Ren Yan, Ming-Te Chi, Tong-Yee Lee, and Wen-Chieh Lin. 2008. Stylized Rendering Using Samples of a Painted Image. *IEEE Transactions on Visualization and Computer Graphics* 14, 2 (2008), 468–480. doi:10.1109/TVCG.2007.70440

Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. 2023. Rerender A Video: Zero-Shot Text-Guided Video-to-Video Translation. In *SIGGRAPH Asia 2023 Conference Papers* (, Sydney, NSW, Australia,) *(SA '23)*. Association for Computing Machinery, New York, NY, USA, Article 95, 11 pages. doi:10.1145/3610548.3618160

Yonghao Yue, Kei Iwasaki, Bing-Yu Chen, Yoshinori Dobashi, and Tomoyuki Nishita. 2010. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. *ACM Trans. Graph.* 29, 6, Article 177 (dec 2010), 8 pages. doi:10.1145/1882261.1866199