Volumetric Temporal Texture for Smoke Stylization using Dynamic Radiance Fields

Dongqing Wang
https://dongqing-wang.com/
Ehsan Pajouheshgar
https://pajouheshgar.github.io/
Yitao Xu
https://xyt0098.github.io/
Tong Zhang
https://people.epfl.ch/tong.zhang
Sabine Süsstrunk
https://people.epfl.ch/sabine.susstrunk

Image and Visual Representation Lab
École Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland

Abstract

The exemplar-based stylization of dynamic 3D volume sequences remains challenging in computer graphics. The difficulty lies not only in maintaining the styling features concerning a reference image across multiviews and time frames, but also in preserving visually plausible motion as the original smoke simulation, within a reasonable time and computational resources. In this work, we introduce **V**olumetric **N**eural **C**ellular **A**utomata (VNCA), a novel method that synthesizes **volumetric temporal textures** for smoke stylizations in real-time. Our method formats radiance fields atop the self-emerging Neural Cellular Automata (NCA) and generates a dynamic sequence of stylized color and density volumes. The synthesized temporal textures align with the input sequence's overall motion patterns while matching the style of a reference image. We use flow-guided supervision to align the texture's temporal motion with that of the perceived smoke sequence, reducing the training time by over an order of magnitude. We demonstrate that VNCA can be easily adapted for mesh stylization, akin to solid texture modeling, extending its application beyond dynamic volume simulations.

1 Introduction

Creating realistic and artistically expressive 3D smoke sequences is a common task in digital film production and is still an active research problem in computer graphics and vision. The nonlinearity of the underlying physical model, described by the Navier–Stokes equations [2], leads to the inherent turbulence patterns in 3D smoke simulations. It is thus difficult to directly stylize a sequence of smoke densities with the desired appearance while preserving visually plausible motion.

Existing methods for smoke stylization generally stylize *individual* density frames given a reference style image and subsequently perform inter-frame smoothing to ensure temporal

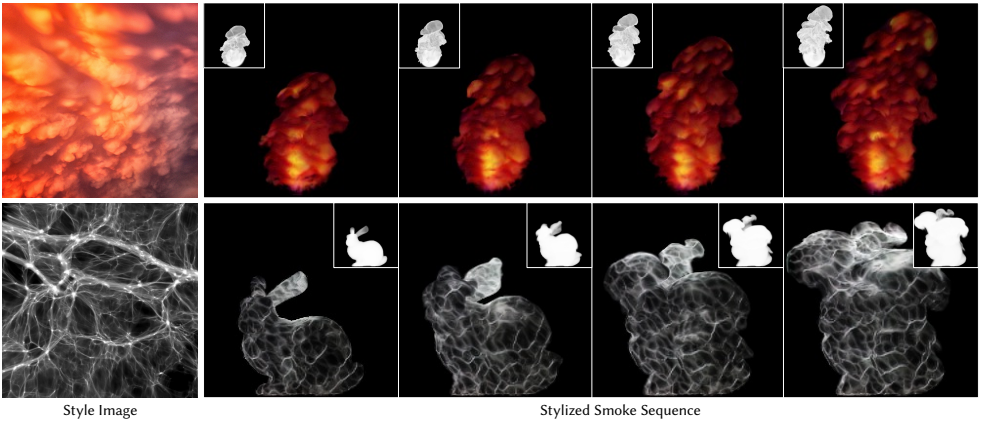

Figure 1: Given a 3D smoke simulation and a style image, we synthesize a temporal texture using Volumetric Neural Cellular Automata (VNCA), aligning stylizing color and density with the style image while preserving the motion of the input sequence.

coherence. Grid-based volumetric Neural Style Transfer (NST) techniques [1, 10] model a velocity field to stylize smoke simulation by per-frame optimization but struggle with inefficient temporal coherency enforcement. Their Lagrangian counterpoint [11] improves inter-frame consistency by tracking per-particle attributes such as position and density given the naturally dynamic structure. However, it requires computationally expensive per-frame conversion from volumetric data to particles, which is impractical for production.

In this work, we propose real-time stylization of 3D volumetric simulations using temporal textures modeled with dynamic radiance fields. Our method is grid-based **and** inherently captures dynamic motion, eliminating the need for costly per-frame optimization and avoiding the overhead of data conversion. The key insights of our work are: (1) synthesizing a dynamic sequence of radiance fields to represent the color and density stylization of 3D temporal textures; and (2) utilizing the inherently dynamic Neural Cellular Automata (NCA) model to host the radiance fields, capturing stylization features while preserving overall visually plausible motion. We term our method **V**olumetric **N**eural **C**ellular **A**utomata (VNCA).

VNCA is parameterized by a small *recurrent* neural network and is fast to train. Trained on a single frame of the density sequence, it can generalize stylization to unseen density fields during inference in *real-time*. VNCA inherits spontaneous moving patterns on its generated textures, which can be supervised by optical-flow [29] using the velocity field from the smoke simulation. This way, we bypass the need for explicit texture advection or interframe smoothing terms and enhance visual plausibility and interframe coherence. These advantages allow us to achieve a training speedup of over an order of magnitude while creating high-quality smoke stylization in real time. Moreover, we show that VNCA can be adapted to mesh stylization by utilizing a slice-based training strategy similar to Solid Texturing methods, achieving results on par with existing works [8, 13, 23]. We provide experiments on both smoke simulation and meshes to showcase our results.

To summarize, our contributions are as follows:

- We present VNCA, a novel volumetric temporal texture model for real-time stylization of 3D smoke simulations modeled by dynamic radiance fields.
- VNCA speed up baseline training by over an order of magnitude.
- VNCA can be generalized to mesh stylization.

2 Related Works

Temporal Texture Synthesis for 3D Smoke Stylization. Existing research on 3D smoke stylization models the volume by advecting density propagated by the velocity and transfers 2D image semantics structure [10] through optimization. Several follow-up works subsequently enable color transfer on 2D [3], improve inter-frame consistency [11], and reduce long training time [1]. However, the optimizations remain computationally heavy because of simulation-based advection. Guo et al. [7] and Aurand et al. [1] train 3D CNNs to bypass iterative optimizations and directly output stylized albedo volume; however, they still require over 18 hours of training. To propose a naturally dynamic stylization technique that overcomes the complexity for temporal smoothing, we draw inspiration from temporal texture synthesis, which can reproduce motion patterns with the same motion distribution as a source texture [14, 25]. This technique is often used to synthesize realistic motion for natural phenomena such as fire and smoke in 2D [27, 28, 30].

Inspired by 2D temporal texture synthesis, our approach to smoke stylization utilizes modeling of temporal texture on 3D voxels for innate temporal consistency. The synthesized volumetric temporal texture is guided by the velocity field of the input density sequence, inspired by prior work on flow-guided texture synthesis [9, 14]. To the best of our knowledge, we are the first to stylize 3D volume with temporal textures.

Texture Synthesis with Neural Cellular Automata. Cellular Automata (CA) consist of regular grids of cells that are updated based on their neighboring cell states. All cells follow the same update rule asynchronously [17], allowing strong generalizability to unseen situations. Gilpin [6] parameterizes the update rule of CA with lightweight convolutional neural networks, creating the foundation for Neural Cellular Automata (NCA) models. Niklasson et al. [19] use NCA to synthesize 2D textures with self-emerging motion using static images as exemplars. Pajouheshgar et al. [21] modify the NCA model for controllable 2D dynamic texture through a pre-trained optical-flow network [29] and vector field supervision. Pajouheshgar et al. [22] generalizes the NCA models to enable temporal textures on meshes. Despite the improvements to the NCA model for texture synthesis, none combine its self-emerging motions with realistic motion synthesis for natural phenomena.

We hereby introduce a variation of NCA on volume grids as a novel presentation for volumetric temporal textures. VNCA synthesizes a dynamic sequence of radiance fields for stylization with visually-plausible motion that naturally emerges post-training. Additionally, VNCA, conditioned on a specific density field during training, can stylize unseen density fields at inference time.

3 Method

3.1 Background: Neural Cellular Automata

NCA operates on a 2D grid of cells with size $H \times W$, and each cell stores its state information in a vector with C channels [17]. The first 3 channels are interpreted as R, G, and B values, and the cell states are initialized with zero values. At each time step, the cell state updates once. $\mathbf{S}(t) \in \mathbb{R}^{H \times W \times C}$ denotes the *cell states* at time t and $s_{ij}(t) = \mathbf{S}(\mathbf{p}, t) \in \mathbb{R}^C$ represents the cell state at $\mathbf{p} = (i, j)$. The *update rule* is a trainable PDE that determines how cell state changes over time depending on $s_{ij}(t)$ and its spatial gradients and Laplacian. All cells share the same local update rule with *two steps*. First, each cell state convolves a set of filters over its surrounding cells to form a *perception vector* containing neighborhood

information. Then, the perception vector is passed through two Fully Connected layers and a random binary mask to determine the new cell state.

3.2 Volumetric NCA Representation

We define the Volumetric NCA grids on a 3D voxel-based representation. To avoid confusion, we refer to the voxels in VNCA as "cells", and those in the input density field as "voxels". We map $t = N$ steps of cell state update to two adjacent density frames for an approximated motion for smoke at inference time.

Our VNCA model is applied to each density frame $d : \mathbb{R}^3 \to \mathbb{R}$ by synthesizing the radiance field $\{R, G, B, \Delta d\}$ for stylization. We construct our VNCA through a voxelized cube of resolution $H \times W \times D$. The cell state at time t is thus denoted as $\mathbf{S}(t) \in \mathbb{R}^{H \times W \times D \times C}$, with $s_{ijk}(t) \in \mathbb{R}^C$ as a vector denoting the C-channel cell state at (i, j, k). The R, G, and B values for color stylization and Δd as the density stylization to be applied to each corresponding voxel will be decoded from the cell state for stylization, detailed in the supplementary. All channels are initialized with 0s.

3.3 Update Rule

The update rule is a partial differential equation (PDE) on cell states: $\mathbf{S}(\mathbf{p}, t + \Delta t) = \mathbf{S}(\mathbf{p}, t) + \frac{\partial \mathbf{S}(\mathbf{p},\mathbf{t})}{\partial t} \Delta t$, where $\mathbf{p}$ is the spatial location and t is the time step. We solve this PDE by discretizing the 3D space using a voxelized grid and applying the update rule with fixed-sized time steps. Figure 2 illustrates the update rule along with the VNCA cell state.

Perception vector. To perceive information from all neighborhood cells around a given voxel, we apply 3D convolutions with a set of filters to the cell states for perception vectors. The filters are Sobel operators and the 27-point variant of the discrete Laplace operator [20] with frozen weights during training. The perception vector $z_{ijk} \in \mathbb{R}^{5C}$ for s_{ijk} is thus given by the concatenation of the cell state, the discretized gradient, and the laplacian: $z_{ijk} = s_{ijk} \parallel (\partial_x \mathbf{S}|_{ijk}) \parallel (\partial_y \mathbf{S}|_{ijk}) \parallel (\partial_z \mathbf{S}|_{ijk}) \parallel (\nabla^2 \mathbf{S}|_{ijk})$.

Encoding Priors. The perception vector is designed to update the cell states towards an aligned style and visually plausible motion. Therefore, the perception vector should be aware of essential information about the stylized objective.

Specifically, we apply a 3-dimensional *positional encoding* [21], a *density encoding*, and a *velocity encoding* as shown in Figure 2. Our positional encoding tensor has three channels and is defined as: $\mathcal{P}_{ijk} = \left[\frac{2i+1}{H} - 1, \frac{2j+1}{W} - 1, \frac{2k+1}{D} - 1\right]$. Using positional encoding improves the stability and consistency of the synthesized motion. For density encoding, we append the 1-channel input density field $\mathcal{D}$ of the current smoke frame in training to the end of the perception vector to allow VNCA to be aware of the structure of the input density. Density encoding encourages style features that closely match the input reference, as opposed to a "perforated" appearance. The velocity encoding $\mathcal{V}$ interpolates the update for the desired advected density field. It provides spatial inductive bias to help VNCA better match the advection of the input sequence and produce visually plausible motion. We provide ablation studies in Section 4.2.

Stochastic Update. We pass the encoded perception vector through our update network represented by a Multi-Layer Perceptron (MLP) f_θ with m layers and a ReLU activation to determine the updated state of each cell. Each cell update follows a geomet-

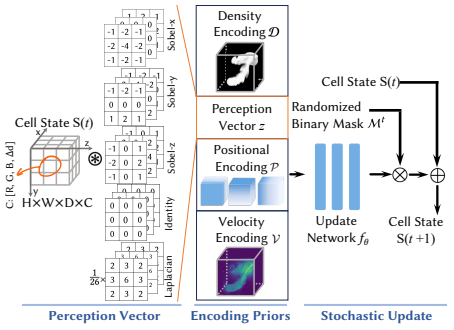

Figure 2: **Update Rule.** The update rule determines the next cell state. We obtain the perception vector with neighbor cell states and concatenate density, velocity, and positional encoding. The stochastic update processes this vector for an update to each cell.

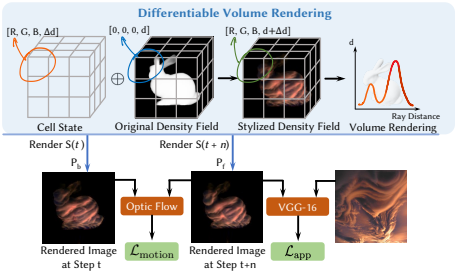

Figure 3: **Training.** At each training epoch, we use a single input frame for the *Density Encoding*, apply the VNCA update rule for n steps, and record the cell state before and after the update. We then render the smoke stylized with cell state $[\mathtt{R},\mathtt{G},\mathtt{B},\triangle\mathtt{d}]$ before and after the updates to get P_b and P_f.

ric process [15] independently from the other cells. This asynchronous cell update is implemented by applying a randomized binary mask $\mathcal{M}^t$ to the output of the MLP: $\frac{\partial s_{ijk}}{\partial t} = \mathcal{M}^t(f_\theta(z_{ijk} \parallel \mathcal{P}_{ijk} \parallel \mathcal{D}_{ijk} \parallel \mathcal{V}_{ijk}))$. Each voxel in the random binary mask $\mathcal{M}$ is a Bernoulli random variable with probability $p = 0.5$. This asynchronous cell update scheme improves the robustness and stability of VNCA [18] and allows the stylization of longer sequences.

3.4 Training for Texturing Smoke Data

We represent the volumetric temporal texture through a trained VNCA. This allows for real-time stylization that aligns with both the reference image and the motion of the input smoke sequence. Figure 3 shows the outline of our training pipeline.

3.4.1 Appearance Supervision

We render the stylized volume by applying the radiance field stylization and rendering with a simplified differentiable volume rendering algorithm [4, 16], which we detail in the supplementary materials. After obtaining the rendered image P_f, we use a pre-trained VGG16 network [26] to extract the style information using deep feature representations [5]. The local cell communication in the update rule of VNCA is shared by all cells and the rendered 2D image involves contributions from all cells. Given this inductive bias, we only use one randomly selected camera position to render the view at each training epoch. This is sufficient for the converged VNCA to match the reference image from all views, contributing to our lower training time over baseline methods.

Our appearance loss is based on the style loss proposed by Kolkin et al. [12] and consists of style-matching and moment-matching terms. We extract two sets of VGG16 feature maps F^r and F^Ψ of size $C' \times H' \times W'$ from the rendered image and the target style image, respectively. C' and $H' \times W'$ represent the number of channels and the spatial dimensions of the VGG16 feature maps, respectively. These feature maps are then flattened along the spatial dimensions to obtain the feature sets A^r and A^Ψ. We measure the style distance D_c between features through the minimum cost of transporting from one set to another using cosine distance:

$$d_c(x,y) = 1 - \frac{x \cdot y}{\|x\|_2 \|y\|_2}, \quad D_c(A^r, A^\Psi) = \frac{1}{H' \times W'} \sum_i \min_j d_c(A_i^r, A_j^\Psi). \tag{1}$$

The subscript indicates the feature vector index in the feature set.

To encourage density changes in the stylization, we additionally render the stylized smoke in grayscale. We then extract deep feature maps F^r_{gray} and F^Ψ_{gray} and feature sets A^r_{gray} and A^Ψ_{gray} from the rendered image and the grayscale stylized image as described above, and similarly define a distance for density pattern matching.

Our final style-matching term $\mathcal{L}_{\text{style}}$ combines the style distance between the grayscale and colored images and is defined as:

$$\overleftrightarrow{D_c}(X,Y) = \max(D_c(X,Y), D_c(Y,X)), \quad \mathcal{L}_{\text{style}} = \overleftrightarrow{D_c}(A^r, A^\Psi) + \overleftrightarrow{D_c}(A^r_{\text{gray}}, A^\Psi_{\text{gray}}). \tag{2}$$

We additionally include a moment-matching term $\mathcal{L}_{\text{moment}}$ to align the magnitudes between colored features A^r and A^Ψ, and between grayscale features A^r_{gray} and A^Ψ_{gray}:

$$D_k(X,Y) = \frac{1}{C'} \|\mu_X - \mu_Y\|_1 + \frac{1}{C'^2} \|\Sigma_X - \Sigma_Y\|_1, \quad \mathcal{L}_{\text{moment}} = D_k(A^r, A^\Psi) + D_k(A^r_{\text{gray}}, A^\Psi_{\text{gray}}). \tag{3}$$

where μ_X and Σ_X denote the mean and covariance of the feature set X. Our overall appearance loss is the sum of the style-matching and moment-matching terms: $\mathcal{L}_{\text{app}} = \mathcal{L}_{\text{style}} + \mathcal{L}_{\text{moment}}$. In our experiments, we utilize deep features at different scales and sum our appearance loss over different layers of VGG16.

3.4.2 Flow-guided Motion Supervision

Instead of explicitly simulating transport, we supervise the emerged motion of the synthesized temporal texture with the input velocity field to align with the smoke movement. Since a pre-trained model for estimating 3D motion vectors does not exist, it is costly and unfeasible to directly compare the 3D smoke motion and our volumetric temporal texture.

We project the velocity field of input smoke data onto a 2D camera plane with camera pose Φ and denote the projection as $\mathcal{J}^\Psi \in \mathbb{R}^{H \times W \times 2}$ to supervise the texture motion. The motion of our temporal texture can be quantified by comparing the variation of rendered views in adjacent time frames with camera pose Φ. Specifically, let P_b and P_f be two rendered images of the stylized density field with VNCA from Φ during training, with P_b as the image rendered before a random n update steps is applied to VNCA, and P_f as the image after the updates. We use a pre-trained optical flow prediction network $\mathcal{F}_{\text{OF}}$ [29] to predict the perceived motion between P_b and P_f given by $\mathcal{J}^r = \mathcal{F}_{\text{OF}}(P_b, P_f)$ as 2D vector. We align the directions of the target vector field and the predicted vector by computing the cosine distance given by $\mathcal{L}_{dir}$:

$$\mathcal{L}_{\text{dir}} = \frac{1}{H \cdot W} \sum_{i,j} d_c(\mathcal{J}^r_{ij}, \mathcal{J}^\Psi_{ij}). \tag{4}$$

To obtain a texture motion that has roughly the same magnitude as the input vector field, we normalize the predicted vector field with the number of update steps n between P_b and P_f and the N update steps we use between each pair of adjacent rendered views:

$$\mathcal{L}_{\text{mag}} = \frac{1}{H \cdot W} \sum_{i,j} \left| \frac{N}{n} \cdot \left\|\mathcal{J}^r_{ij}\right\|_2 - \left\|\mathcal{J}^\Psi_{ij}\right\|_2 \right|. \tag{5}$$

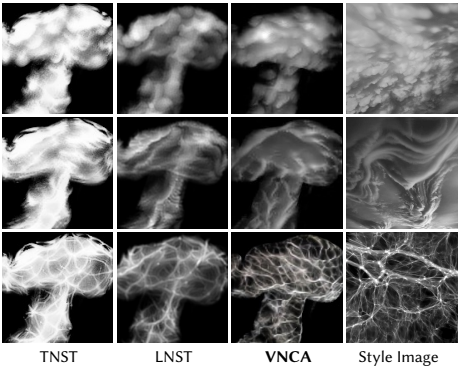

Figure 4: **Qualitative Stylization Comparison with TNST and LNST.** VNCA synthesizes stylization that closely matches the reference image, with superior quality indicated by our user study.

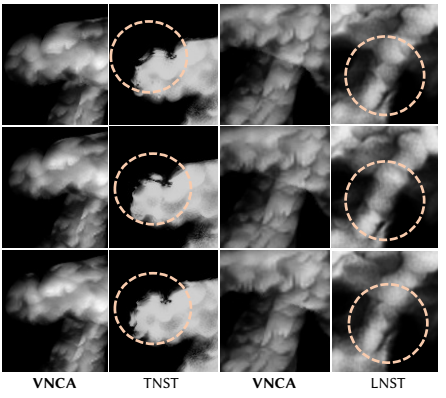

Figure 5: **Between-frame Coherence Comparison with baseline.** VNCA stylization is more coherent across frames compared to our baseline methods.

The final motion supervision loss $\mathcal{L}_{motion}$ is given by: $\mathcal{L}_{\text{motion}} = \max(0, \mathcal{L}_{\text{dir}} - 1) \cdot \lambda_{\text{mag}} \mathcal{L}_{\text{mag}} + \lambda_{\text{dir}} \mathcal{L}_{\text{dir}}$, such that the training prioritizes direction alignment between the target and predicted motion over their magnitudes.

At each epoch, we condition one density field to ensure the optical flow network is unbiased by the motion of the input density sequence. We rotate the camera position between epochs for multiview flow-guided supervision of texture motion. For inference, we apply N update steps on cell states for a visible motion on each pair of adjacent density fields.

4 Experiments

4.1 Comparison with Prior Works

Baseline Methods. We compare VNCA with 3D smoke stylization methods Kim et al. [10] and Kim et al. [11]. We additionally qualitatively compare with Guo et al. [2] and Aurand et al. [1] on their reported results and include the comparison in the supplementary materials.

Stylization Comparison. Our work focuses on 3D volumetric temporal texture. Since TNST and LNST do not support 3D color stylization, we use grayscale reference images for all methods, We run experiments on the same RTX3090Ti for fairness. Figure 4 shows the qualitative comparison of stylization matching quality. VNCA can synthesize features that match the reference pattern and turbulence effect on par with TNST and LNST.

Figure 5 compares the preservation of stylization coherence between frames. We highlight the circled regions for inconsistent stylization between consecutive frames in prior works. As they stylize each density field independently, there are inevitable inter-frame inconsistencies. In contrast, the volumetric temporal texture synthesized by VNCA is naturally dynamic from the emerging motion of the underlying NCA model.

Table 1 shows the results of our user study on the stylization quality, inter-frame consistency, and preservation of smoke motion comparison for TNST, LNST, and VNCA. We

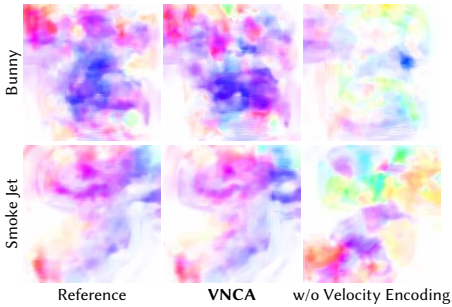

Figure 6: **Ablation on Velocity Encoding.** VE better aligns the estimated optical flow of our synthesized texture motion with the input velocity field.

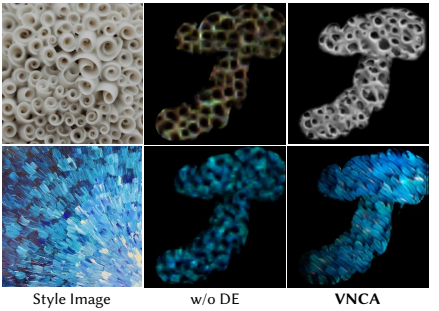

Figure 7: **Ablation Studies on Density Encoding.** Without density encoding, the synthesized appearance is porous and inconsistent with the input reference.

Method	Runtime	Stylization Quality	Inter-frame Consistency	Motion Preservation
TNST	463s	0.67%	n/a	16.33%
LNST	10s	19.33%	15.91%	28.57%
VNCA	0.8s	80%	84.09%	55.10%

Table 1: **Quantitative Comparison with TNST and LNST.** The runtime for prior works is evaluated **without omni-view training** on an RTX3090Ti.

Loss	Positional Encoding	**P + V Encoding**	Loss	w/o Density Encoding	**Density Encoding**
$\mathcal{L}_{\text{dir}}$	0.134	0.083	$\mathcal{L}_{\text{app}}$	7.29	5.42
$\mathcal{L}_{\text{mag}}$	0.295	0.167	-		

Table 2: **Quantitative Ablation on the Encoded Priors.** By encoding the density field and velocity field of the smoke sequence, the synthesized temporal texture of VNCA achieves better appearance and motion alignment with the stylizing target.

received 50 valid user responses for stylization quality evaluation, and 49 for motion preservation evaluation. TNST is not compared for inter-frame coherence as LNST is shown to be better [11]. Each entry shows the percentage of users who prefer the stylization of the corresponding methods. We see a preference for our methods over the baseline methods, and we refer to the supplementary material for details and more qualitative results.

Runtime Comparison. In Table 1, we provide a quantitative comparison with TNST and LNST on the time required to stylize one frame of the input sequence *Smoke Jet* dataset ($200 \times 300 \times 200$) with publicly available code on the required per-frame stylization time and conduct a user study to compare the inter-frame consistency and stylization quality. We can see that our average inference time for a single frame is faster than prior methods, while our training already includes omniview consistency.

4.2 Ablation Studies

We evaluate the design choice of VNCA on encoding priors and loss weights in our ablation studies. Ablation on our multiview training paradigm is in the supplementary.

Density encoding. By appending one of the input density frames to the perception vector, we provide VNCA with the global density distribution in the input data. This information is crucial for appearance stylization since each cell in the NCA's cell state is only aware of its immediate neighborhood. As shown in Figure 7, the stylization without density encoding results in an overall color that is somewhat close to the reference but is abundantly porous and lacks detailed structures, such as stroke patterns. Including density encoding significantly

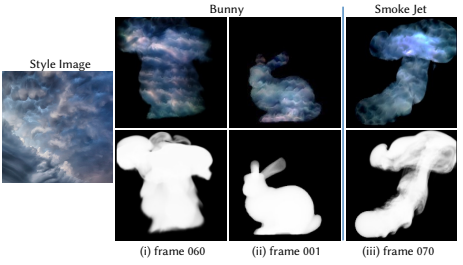

Figure 8: **Generalization Ability of VNCA.** During inference, the trained VNCA can stylize the full sequence of *Bunny* and *Smoke Jet* unseen in training.

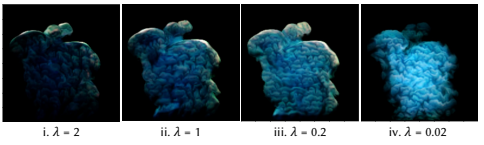

Figure 9: **Comparison of different weights for loss functions.** $\lambda = \lambda_{\text{motion}}/\lambda_{\text{app}}$. When the weight for the motion loss is too high w.r.t appearance loss, the stylization degenerates.

enhances the quality of stylization in terms of matching the given style image.

Velocity encoding. Velocity encoding facilitates the alignment of our volumetric temporal texture and guiding velocity field. We compare the estimated optical flow between the synthesized texture motion and the projected velocity field on the smoke datasets in Figure 6. The synthesized result with velocity encoding shows a closer alignment to the guiding motion flow. Table 2 shows a quantitative comparison of the impact of velocity encoding on the alignment of motion direction and magnitude, and the impact of density encoding on the appearance supervision.

Comparison on weights for loss functions Figure 9 shows the ratio $\lambda = \lambda_{\text{motion}}/\lambda_{\text{app}}$ between the weight of $\mathcal{L}_{\text{motion}}$ to that of $\mathcal{L}_{\text{app}}$. The stylized appearance can be corrupted when the weight for the motion loss is too high compared to the appearance loss.

4.3 Generalization Ability

A key feature of VNCA is its robust generalizability as an NCA model. At each epoch, we condition VNCA on the current density field of the training frame. As we condition the density field to be stylized during inference and supply the velocity field, VNCA does not require re-training to synthesize matching appearance and corresponding visually plausible motion. This way, a converged volumetric temporal texture can stylize the entire density sequence. We empirically find that trained VNCA can even extrapolate to smoke datasets unseen at training time. In Figure 8, the trained model is conditioned on the 60th frame of the *bunny* dataset in (i). At inference time, the trained texture volume (ii) can stylize all other frames of the bunny scene. Additionally, (iii) it is capable of stylizing unseen *Smoke Jet*. This versatility allows our model to reduce training time compared to baseline methods.

4.4 Application to Mesh Texturing

VNCA can easily be extended for mesh texturing, given a reference style following the line of research in Solid Texture Synthesis [8, 13, 24]. To train our model for mesh texturing, we disable the motion loss and apply our appearance loss, $\mathcal{L}_{\text{app}}$, on random axis-aligned slices of the synthesized texture block.

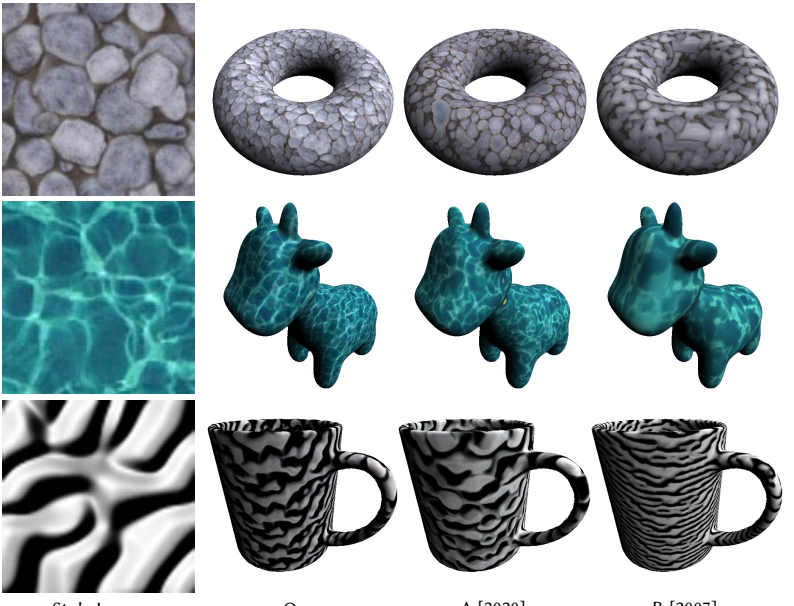

Figure 10: **Stylization on Meshes.** VNCA can be generalized and applied to mesh texturing. A. Gutierrez et al. [8], B. Kopf et al. [13]. Our method synthesizes mesh stylization with better consistency with the style image and fine-level details.

Figure 10 shows the stylized meshes by our method in comparison with the results from A. Gutierrez et al. [8] and B. Kopf et al. [13] as two representative solid texturing methods for five different style images. As can be seen on the pebble stone texture, the results from both A and B have unclosed pebbles, while our synthesized texture displays complete stones. VNCA also synthesizes textures that are more consistent with the spatial layout of input textures. Despite not being explicitly designed for solid texturing, our method achieves results on par with or better than the previous solid texturing methods. More results are provided in the supplementary.

5 Conclusion

In this paper, we present VNCA, a novel temporal texture for stylizing volumetric smoke sequences using 2D exemplars. Our method achieves 1. high-quality appearance stylization with multiview and across-frame coherence w.r.t a reference image, 2. visually plausible motion as the input, and 3. speedup training time. Our model can also be extended to mesh texturing.

Limitations Since we use the NCA framework, we inherit its shortcomings in that the homogeneous update rule for NCA limits the synthesized texture from achieving high heterogeneity. VNCA can extract repeated patterns in the style image as features for stylization, but may not be able to synthesize stylizing features with a clear foreground and background.

Acknowledgement. This work was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5-180359. The authors thank Ziyi Zhang, Martin Everaert, Corentin Dumery, and Michele Vidulis for their time spent proofreading and for their kind suggestions during the paper writing.

References

[1] Joshua Aurand, Raphael Ortiz, Silvia Nauer, and Vinicius C Azevedo. Efficient neural style transfer for volumetric simulations. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(6):1–10, 2022.

[2] Robert Bridson. *Fluid Simulation for Computer Graphics*. CRC press, 2015.

[3] Fabienne Christen, Byungsoo Kim, Vinicius C Azevedo, and Barbara Solenthaler. Neural smoke stylization with color transfer. *arXiv preprint arXiv:1912.08757*, 2019.

[4] Julian Fong, Magnus Wrenninge, Christopher Kulla, and Ralf Habel. Production volume rendering: Siggraph 2017 course. In *ACM SIGGRAPH 2017 Courses*, pages 1–79. 2017.

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[6] William Gilpin. Cellular automata as convolutional neural networks. *Physical Review E*, 100(3):032402, 2019.

[7] Jie Guo, Mengtian Li, Zijing Zong, Yuntao Liu, Jingwu He, Yanwen Guo, and Ling-Qi Yan. Volumetric appearance stylization with stylizing kernel prediction network. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 40(4):162–1, 2021.

[8] Jorge Gutierrez, Julien Rabin, Bruno Galerne, and Thomas Hurtut. On demand solid texture synthesis using deep 3d networks. In *Computer Graphics Forum*, volume 39, pages 511–530. Wiley Online Library, 2020.

[9] Ondřej Jamriška, Jakub Fišer, Paul Asente, Jingwan Lu, Eli Shechtman, and Daniel Sỳkora. Lazyfluids: Appearance transfer for fluid animations. *ACM Transactions on Graphics (TOG)*, 34(4):1–10, 2015.

[10] Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. Transport-Based Neural Style Transfer for Smoke Simulations. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 38(6):188, 2019.

[11] Byungsoo Kim, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. Lagrangian neural style transfer for fluids. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 39(4), aug 2020.

[12] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10051–10060, 2019.

[13] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, Oliver Deussen, Dani Lischinski, and Tien-Tsin Wong. Solid texture synthesis from 2d exemplars. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):2:1–2:9, 2007.

[14] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24(3):795–802, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073263. URL https://doi.org/10.1145/1073204.1073263.

[15] Ye (Lam Yeh) Lin. Geometric processes and replacement problem. *Acta Mathematicae Applicatae Sinica*, 4(4):366–377, Nov 1988. ISSN 1618-3932. doi: 10.1007/BF02007241. URL https://doi.org/10.1007/BF02007241.

[16] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *The 16th European Conference on Computer Vision ECCV 2020*, 2020.

[17] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 2020. doi: 10.23915/distill.00023. https://distill.pub/2020/growing-ca.

[18] Eyvind Niklasson, Alexander Mordvintsev, and Ettore Randazzo. Asynchronicity in neural cellular automata. In *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press, 2021.

[19] Eyvind Niklasson, Alexander Mordvintsev, Ettore Randazzo, and Michael Levin. Self-organising textures. *Distill*, 2021. doi: 10.23915/distill.00027.003. https://distill.pub/selforg/2021/textures.

[20] Randall C. O'Reilly, Jeffrey M. Beck, C. R., and O'REILLY And. A family of large-stencil discrete laplacian approximations in three dimensions. In *Int. J. Numer. Methods Eng*, 2006. URL https://api.semanticscholar.org/CorpusID:10517137.

[21] Ehsan Pajouheshgar, Yitao Xu, Tong Zhang, and Sabine Süsstrunk. Dynca: Real-time dynamic texture synthesis using neural cellular automata. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20742–20751, 2023.

[22] Ehsan Pajouheshgar, Yitao Xu, Alexander Mordvintsev, Eyvind Niklasson, Tong Zhang, and Sabine Süsstrunk. Mesh neural cellular automata. *ACM Transactions on Graphics (TOG)*, 43(4):1–16, 2024.

[23] Darwyn R Peachey. Solid texturing of complex surfaces. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 279–286, 1985.

[24] Ken Perlin. An image synthesizer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 19(3):287–296, 1985.

[25] Ashfaqur Rahman and Manzur Murshed. A motion-based approach for temporal texturessynthesis. In *TENCON 2005-2005 IEEE Region 10 Conference*, pages 1–4. IEEE, 2005.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#SimonyanZ14a.

[27] Jos Stam and Eugene Fiume. Depicting fire and other gaseous phenomena using diffusion processes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 129–136, 1995.

[28] Martin Szummer and Rosalind W Picard. Temporal texture modeling. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 823–826. IEEE, 1996.

[29] Matthew Tesfaldet, Marcus A Brubaker, and Konstantinos G Derpanis. Two-stream convolutional networks for dynamic texture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6703–6712, 2018.

[30] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 479–488, 2000.