SNeRF: Stylized Neural Implicit Representations for 3D Scenes

THU NGUYEN-PHUOC, FENG LIU, and LEI XIAO, Reality Labs Research, Meta, USA

Fig. 1. Given a neural implicit scene representation trained with multiple views of a scene, SNeRF stylizes the 3D scene to match a reference style. SNeRF works with a variety of scene types (indoor, outdoor, 4D dynamic avatar) and generates novel views with cross-view consistency.

This paper presents a stylized novel view synthesis method. Applying state-of-the-art stylization methods to novel views frame by frame often causes jittering artifacts due to the lack of cross-view consistency. Therefore, this paper investigates 3D scene stylization that provides a strong inductive bias for consistent novel view synthesis. Specifically, we adopt the emerging neural radiance fields (NeRF) as our choice of 3D scene representation for their capability to render high-quality novel views for a variety of scenes. However, as rendering a novel view from a NeRF requires a large number of samples, training a stylized NeRF requires a large amount of GPU memory that goes beyond an off-the-shelf GPU capacity. We introduce a new training method to address this problem by alternating the NeRF and stylization optimization steps. Such a method enables us to make full use of our hardware memory capacity to both generate images at higher resolution and adopt more expressive image style transfer methods. Our experiments show that our method produces stylized NeRFs for a wide range of content, including indoor, outdoor and dynamic scenes, and synthesizes high-quality novel views with cross-view consistency.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Image-based rendering**; **Non-photorealistic rendering**.

Additional Key Words and Phrases: neural style transfer, implicit scene representations, view synthesis, stylization

Authors' address: Thu Nguyen-Phuoc, thunp@fb.com; Feng Liu, fliu@cs.pdx.edu; Lei Xiao, lei.xiao@fb.com, Reality Labs Research, Meta, USA.

1 INTRODUCTION

With the increasing availability of new social media platforms and display devices, there has been a growing demand for new visual 3D content, ranging from games and movies to applications for virtual reality (VR) and mixed reality (MR). In this paper, we focus on the problem of stylizing 3D scenes to match a reference style image. Imagine putting on a VR headset and walking around a 3D scene: one is no longer constrained by the look of the real world, but instead can view how the world would look like through the artistic lenses of Pablo Picasso or Claude Monet.

Naively applying image-based stylization techniques [Gatys et al. 2016] to 3D scenes might lead to flickering artefacts between different views, since each view is stylized independently without any consideration for the underlying 3D structure. Therefore, recent work has explored various choices of 3D representations to address this issue: one can stylize the underlying 3D scenes and then render new consistent views from them [Huang et al. 2021; Kopanas et al. 2021]. However, these methods does not capture the target style well since they only stylize the scene's appearance, although geometry is also an important part of styles [Kim et al. 2020; Liu et al. 2021].

Recently, neural radiance fields (NeRF) [Mildenhall et al. 2020] offers a compact 3D scene representation that produces high-quality novel-view synthesis results. Later work shows the flexibility of NeRF as 3D scene representations, ranging from large outdoor scenes [Zhang et al. 2020] to dynamic avatars [Gafni et al. 2021]. Its

compactness, expressiveness and flexibility make NeRF an attractive choice of 3D representation for stylization. However, adopting NeRF for neural style transfer poses a great memory constraint. To render a pixel from NeRF, one has to sample densely along a camera ray. This requires high memory usage for rendering and performing back-propagation (for example, it takes 17, 934 MB to render an image patch of size 81×67 [Chiang et al. 2022]). Concurrent work by Chiang et al. [2022] addresses this limitation by performing stylization on rendered patches of NeRF instead of the whole images. However, results from patch-based approaches tend to suffer from global style inconsistencies due to the mismatch between patches in the target style and content images [Huang and Belongie 2017].

We propose to combine NeRF and image-based neural style transfer to perform 3D scene stylization. While NeRF provides a strong inductive bias to maintain multi-view consistency, neural style transfer enables a flexible stylization approach that does not require dedicated example inputs from professional artists [Fišer et al. 2016; Sýkora et al. 2019a; Texler et al. 2020]. Additionally, we address the memory limitations of NeRF by splitting the 3D scene style transfer process into two steps that run alternatingly. This enables us to fully utilize the memory capacity of our hardware to either render NeRF or perform neural style transfer on images with high resolutions.

In this paper, we present SNeRF, a 3D scene neural stylization framework that generates novel views of a stylised 3D scene while maintaining cross-view consistency. Our primary technical contributions include the following:

- We introduce a novel style transfer algorithm with neural *implicit* 3D scene representations, producing high-quality results with cross-view consistency.
- We introduce a general, plug-and-play framework, where various implicit scene representations and stylization methods can be plugged in as a sub-module, enabling results on a variety of scenes: indoor scenes, outdoor scenes and 4D dynamic avatars.
- We develop a novel training scheme to effectively reduce the GPU memory requirement during training, enabling high-resolution results on a single modern GPU.
- Through both objective and subjective evaluations, we demonstrate that our method delivers better image and video quality than state-of-the-art methods.

2 RELATED WORK

2.1 Image and video style transfer

Style transfer aims to synthesize an output image that matches a given content image and a reference style image. Image analogies by Hertzmann *et al.* [2001] and follow-up work [Liao et al. 2017] address this problem by finding semantically-meaningful dense correspondences between the input images, which allows effective visual attribute transfer. However, they require the content and style image to be semantically similar. Stylize-by-example approaches also adopt a patch-based approach using high-quality examples as guidance [Fišer et al. 2016; Sýkora et al. 2019b; Texler et al. 2019]. Despite impressive results, these methods require dedicated guiding examples provided by professional artists. A blind approach to image stylization, neural style transfer, is later on proposed by Gatys et al.

[2016]. Unlike example-based approaches, neural style transfer can perform stylization on arbitrary style reference images. Originally, this is done by optimizing the output image to match the statistics of the content and style images, which are computed using a pre-trained deep network. This optimization process is later replaced by feed-forward networks to speed up the stylization process [Johnson et al. 2016; Ulyanov et al. 2016]. Instead of redoing the stylization for every new style, recent frameworks use the adaptive instance normalization (AdaIN) [Huang and Belongie 2017], whitening and coloring transform (WCT) [Li et al. 2017], linear transformation (LST) [Li et al. 2019], or feature alignment [Svoboda et al. 2020] to perform style transfer with arbitrary new styles at test time.

While relatively similar to image style transfer, video style transfer methods mainly focus on addressing the temporal consistency across the video footage. Recently, key-frame based approaches [Chiang et al. 2022; Jamriška et al. 2019] expand stylize-by-examples to videos and have shown impressive results, but require guiding examples from artists for every keyframe. For blind approaches without dedicated guiding style reference, this can be done using optical flow to calculate temporal losses [Chen et al. 2017, 2020] or align intermediate feature representations [Gao et al. 2018; Huang et al. 2017] to stabilize models' prediction across nearby video frames. Recently, there have been efforts to improve consistency and speed for video style transfer for arbitrary styles through temporal regularization [Wang et al. 2020a], multi-channel correlation [Deng et al. 2021], and bilateral learning [Xia et al. 2021]. Similarly, style transfer for stereo images [Chen et al. 2018; Gong et al. 2018] also aims to achieve cross-view consistency by using dense pixel correspondences (via stereo matching) constraints. However, these methods mostly focus on improving short-range consistency between nearby frames or views, and do not support novel view synthesis.

2.2 3D style transfer

While style transfer in the image domain is a popular and widely studied task, style transfer in the 3D domain remains relatively new. Most approaches focus either on stylizing a single object, using either meshes [Ma et al. 2014] or point clouds [Segu et al. 2020], or material [Nguyen et al. 2012]. Later work focuses on performing style transfer on both geometry and texture [Hauptfleisch et al. 2020; Kato et al. 2018; Yin et al. 2021], but still limited to single objects.

For style transfer at 3D scene level, recent approaches use point clouds [Cao et al. 2020; Huang et al. 2021; Kopanas et al. 2021] or meshes [Höllein et al. 2021] as scene representations. However, these approaches are limited to static scenes. Concurrent work by Chiang et al. [2022] uses implicit scene representations, in particular, NeRF, for 3D scene stylization. However, they only work with static outdoor scenes, while we show that our method works on a variety of scene types. Moreover, due to memory constraints, they only perform stylization on image patches, which tend to generate results that lack global style consistency and thus does not capture the reference style well. Meanwhile, our proposed stylization approach can be trained with images in full resolution and adopt more memory-intensive style transfer approaches such as ArcaneGAN [Spirin 2021]. Finally, their method only focuses on stylizing the

appearance of the scene, although geometry has been acknowledged to be an important factor of style [Kim et al. 2020; Liu et al. 2021].

2.3 Novel view synthesis

Novel view synthesis aims to estimate images at unseen viewpoints from a set of posed source images. When source images can be sampled densely, light field approaches work well [Gortler et al. 1996; Levoy and Hanrahan 1996]. Other approaches often use the scene geometrical proxy to warp and blend input views to create novel views [Buehler et al. 2001; Chaurasia et al. 2013; Penner and Zhang 2017; Zitnick et al. 2004]. Recently, a wide variety of deep learning-based approaches have been developed for novel view synthesis [Flynn et al. 2016; Hedman et al. 2018; Kalantari et al. 2016]. Aliev *et al.* [2020] develop a neural point-based rendering method. This method associates a feature descriptor for each 3D point, projects the feature descriptors to the target view, and finally uses a neural network to synthesize the target view. Deferred neural rendering employs a similar approach to mesh-based rendering [Thies et al. 2019]. Sitzmann *et al.* [2019] develop a scene representation, called DeepVoxels, that can encode the view-dependent effects without explicitly modeling the scene geometry. Zhou *et al.* [2018] estimate multi-plane images from two input images as the scene representation, which can be projected to the target view to render the target view. This method is further improved to take more input views [Mildenhall et al. 2019a], handle larger viewpoint shifts [Srinivasan et al. 2019], and produce VR videos [Broxton et al. 2020]. Mildenhall *et al.* [2020] represent scenes as neural radiance fields (NeRF), and show impressive results for novel view synthesis. Since then, a large number of NeRF methods have been developed [Barron et al. 2021; Li et al. 2021; Martin-Brualla et al. 2021; Zhang et al. 2020], which are described in a survey by Dellaert and Yen-Chen [2021]. Given NeRF's ability to render high-quality views and represent a variety of scene types, we adopt NeRF for stylization to render cross-view consistent stylized novel views.

3 METHOD

Given a 3D scene, we aim to manipulate it such that rendered images of this scene match the style from a reference image I_{style}. Additionally, rendered images of the same scene from different views should be consistent. In this work, we use NeRF as our choice of scene representation for its compactness and flexibility. We propose a memory-efficient training approach that alternates between stylization and NeRF training. This enables us to make full use of our hardware memory for either stylization or training NeRF, but not both at the same time, and thus achieve results with high resolution.

3.1 Preliminaries

3.1.1 NeRF overview. NeRF is a continuous 5D function whose input is a 3D location $\mathbf{x}$ and 2D viewing direction $\mathbf{d}$, and whose output is an emitted color $\mathbf{c} = (r, g, b)$ and volume density σ. NeRF is approximated by a multi-layer perceptron (MLP): $F_{\Theta} : (\mathbf{x}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$, which is trained using the following loss function:

$$L_{NeRF}(F_{\Theta}) = \frac{1}{M} \sum_{i=1}^{M} \left\| \mathbf{c}(\mathbf{r}_i) - \mathbf{c}'(\mathbf{r}_i) \right\|_2 \tag{1}$$

where $\{\mathbf{r}_i\}_{i=1}^{M}$ is a batch of randomly sampled camera rays using the corresponding camera poses and the camera intrinsic at each optimization step, $\mathbf{c}'(\mathbf{r}_i)$ is the color of a pixel rendered from F_{Θ}, and $\mathbf{c}(\mathbf{r}_i)$ is the ground truth pixel color.

In this work, we assume that the stylization process starts with a NeRF pre-trained with realistic RGB images $\{x_i\}_{i=1}^{N}$ and corresponding camera poses $\{\theta_i\}_{i=1}^{N}$. We apply our approach to 3 different NeRF scene types: classic NeRF [Mildenhall et al. 2020] for indoor scenes, NeRF++ [Zhang et al. 2020] for 360°outdoor scenes and finally, dynamic 4D human avatar [Gafni et al. 2021].

3.1.2 Style transfer overview. Given a content image $I_{content}$ and target style reference image I_{style}, we want to generate a new image x' that matches the style of I_{style}, but still maintain the content of $I_{content}$. This is done by optimizing the generated image to match the content statistics of the content image and the style statistics of the style reference image using the following loss functions:

$$L_{Transfer} = L_{Content}(I_{Content}, x') + L_{Style}(I_{Style}, x') \tag{2}$$

$$L_{Content} = \left\| \Phi(I_{Content}) - \Phi(x') \right\|_2 \tag{3}$$

$$L_{Style} = \left\| \Phi(I_{Style}) - \Phi(x') \right\|_2 \tag{4}$$

where Φ are the image statistics, which are usually features extracted from different layers of a pre-trained network such as VGG [Simonyan and Zisserman 2015]. In our case, $I_{content}$ is a rendered view of the scene function, and x' is a stylized version of that view. While most of the results in this paper is stylized using the neural style transfer algorithm proposed by Gatys et al. [2016], we also use a GAN-based method [Spirin 2021] to stylize dynamic 4D avatars.

3.2 Stylizing implicit scene representation

We stylize a 3D scene represented as a NeRF to match a reference style image I_{style} using the following loss function:

$$L_{SNeRF} = L_{NeRF} + L_{Transfer} \tag{5}$$

where $L_{Transfer}$ performs stylization to match a given style image and L_{NeRF} maintains the underlying scene structure to preserve multi-view consistency.

Previous work [Chiang et al. 2022; Huang et al. 2021; Kopanas et al. 2021] optimizes for both losses at the same time to perform scene stylization. This would require rendering full images (or patches as proposed by Chiang et al. [2022]) from NeRF to compute $L_{Transfer}$ at *every training step*, which is time consuming. Additionally, this approach requires that the memory has to be shared between three memory-intensive components: the feature extractor (such as VGG) to compute image statistics, the volumetric renderer of NeRF, and back-propagation. This greatly limits the resolution of stylized results as well as the choice of stylization methods. For example, with a 32GB GPU (NVIDIA V100), we could only perform stylization simultaneously for images at size 252×189 using VGG-16-based losses similar to [Gatys et al. 2016], and quickly ran into OOM error with larger images at size 366×252.

To address the memory burden of the methods described above, we propose an alternating training regime inspired by coordinate descent. Our insight is that we can decouple $L_{Transfer}$ and L_{NeRF}, and minimize one at a time. To compute $L_{Transfer}$, we only need the feature extractor, the target style image, and rendered images
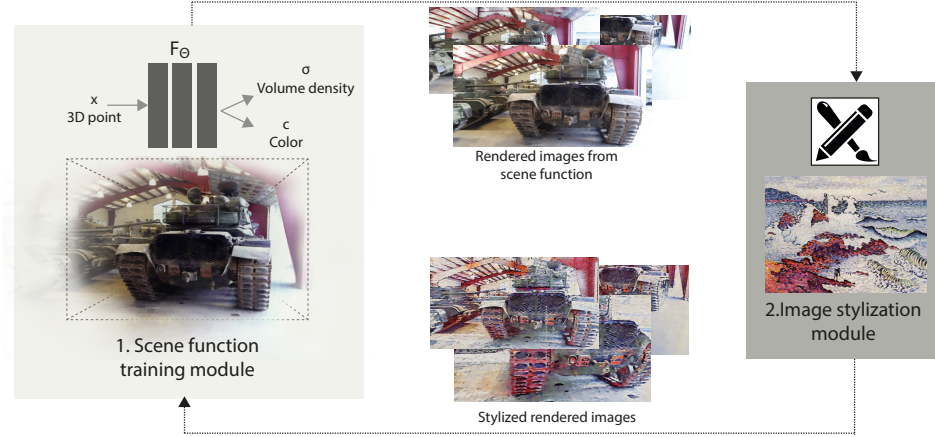
Fig. 2. **Overview**: We propose an alternating training approach to stylize implicit scene representations. For one iteration: (1) Given a pre-trained scene function, we render images from different views. (2) We then stylize these images using the image stylization module. (3) We train the scene function to match multi-view stylized images similar to training a NeRF function. In the next iteration, (1) we again render images from different views from a now more stylized scene function, (2) perform image stylization on this new set of images, and (3) train NeRF with the new set of images.

of the scene, which can be precomputed from NeRF. Meanwhile, to compute L_{NeRF}, we only need the volumetric renderer and target images, which can be precomputed by a separate stylization process. In practice, we train NeRF with L_{NeRF} for a number of steps on batches of randomly sampled rays across different views, before rendering a set of images at different views to compute $L_{Transfer}$. Figure 2 provides an overview of our method.

The proposed alternating training regime allows one to dedicate the full hardware capacity to either image stylization or NeRF training. For image stylization, this enables us to perform stylization on the whole image and achieve more globally consistent stylized results. For NeRF training, we can train NeRF to generate results at higher resolution, and apply our method to dynamic scenes (in particular, dynamic avatar). With the same hardware, our training regime can now stylize NeRF to synthesize images at size 1008×756, 4 times larger than what we previously could when performing training and stylization simultaneously. This also opens up potentials to use more expressive pre-trained models to compute $L_{Transfer}$, such as StyleGAN [Karras et al. 2019] or CLIP [Radford et al. 2021].

3.3 Alternating training regime details

Starting from a set of "realistic" RGB image $\{x_i\}_{i=1}^N$ rendered from pre-trained NeRF using camera poses $\{\theta_i\}_{i=1}^N$, we perform style transfer independently on each image (as the target content) by minimizing $L_{Transfer}$. Note that after this step, the stylized images are not necessarily multi-view consistent. Secondly, we use this set of stylized images $\{x'_i\}_{i=1}^N$ as target images to train the NeRF scene function F_Θ using L_{NeRF}. Note that here we can train NeRF on batches of random rays across multiple views, instead using full images which can be time and memory-consuming. Finally, using the stylised NeRF, we render a new set of images. While these images might not yet capture the full details of the target style, they are multi-view consistent thanks to the underlying scene structure of NeRF. In the next iteration, we perform stylization on

the new set of (more stylized) images of NeRF. By boostraping the image stylization algorithm to the output images of NeRF, we obtain more multi-view consistent stylized results, even when each view is stylized independently. We then use the new set of stylized images to further finetune NeRF. The outline of the overall stylization process is described in Algorithm 1. Please refer to the supplementary video for converging results at each iteration.

ALGORITHM 1: Neural Implicit Scene Representation Stylization

Input: Neural implicit scene function F_Θ pre-trained on realistic multi-view images, target style image I_{Style}

Output: Stylised implicit scene function $\hat{F}_\Theta$.

    Initialize $\hat{F}_\Theta$ with F_Θ.
    for each iteration t = 1,...,T **do**
        Render a set of images $\{x_i\}_{i=1}^K$ using the stylized scene function $\hat{F}_\Theta$.
        Optimize the stylized images $\{x'_i\}_{i=1}^K$ to minimize the style transfer loss: $\sum_{i=1}^K L_{Style}(I_{Style}, x'_i) + L_{Content}(x_i, x'_i)$.
        Optimize $\hat{F}_\Theta$ to minimize $L_{NeRF}(\hat{F}_\Theta)$ using $\{x'_i\}_{i=1}^K$ as reference.

3.4 Implementation Details

For stylization, we use a pre-trained VGG16 network [Simonyan and Zisserman 2015] to extract the image statistics. In particular, we use layer *relu4_1* to extract image features for the content loss, and layers *relu1_1*, *relu2_1*, *relu3_1* and *relu4_1* for the style loss. For the 4D avatar, we use a pre-trained ArcaneGAN model [Spirin 2021].

For each scene, we perform scene stylization for 5 iterations ($T = 5$ in Algorithm 1). For each iteration, we perform neural style transfer optimization for 500 steps for each input image, and train the scene function NeRF for 50000 steps (100000 steps for NeRF++). We use the learning rate of 5e-4 for all of our experiments. We train each model using one NVIDIA V100 GPU.

4 RESULTS

To show the flexibility of our stylization method, we train SNeRF on 3 different scene types: indoor scenes, outdoor scenes and dynamic avatar. For indoor scenes, we train NeRF using scenes *Fern* and *TRex* from the LLFF dataset [Mildenhall et al. 2019b]. For outdoor scenes, we train NeRF++ [Zhang et al. 2020] using scenes *Truck*, *Train*, *M60* and *Playground* from the Tank and Temples dataset [Knapitsch et al. 2017]. We also use our method to stylize 4D avatars [Gafni et al. 2021]. For NeRF scenes, we stylize 3D scenes using images at size 1008×756. For NeRF++, we use images at 980×546 for *Truck*, 982×546 for *Train*, 1077×546 for *M60* and 1008×548 for *Playground*. For the dynamic 4D avatar, we train with images at size 512×512. We use all available training views for image stylization.

In Section 4.1 and 4.2, we compare SNeRF with both 2D and 3D approaches. In particular, we compare our method to the following 4 categories of methods:

- Image stylization → Novel view synthesis: we perform image stylization on the input images and synthesize new views from them using LLFF [Mildenhall et al. 2019b].
- Novel view synthesis → Image stylization: we perform novel view synthesis using the input images and then stylize each new view independently using AdaIN [Huang and Belongie 2017], WCT [Li et al. 2017] and LST [Li et al. 2019].
- Novel view synthesis → Video stylization: we perform novel view synthesis using the input images, compile the results into a video, and then perform video stylization using ReReVST [Wang et al. 2020a] and MCCNet [Deng et al. 2021].
- 3D scene stylization → Novel view synthesis: we compare our method with StyleScene by Huang et al. [2021], a point cloud-based approach and with Chiang et al. [2022], a NeRF-based approach but with a patch-based stylization strategy.

4.1 Qualitative results

We show qualitative comparison with other approaches in Figure 3, in which we compare our approach with image, video and 3D-based approaches. We encourage our readers to look at the supplementary videos to see the full effectiveness of our approach in generating cross-view consistent 3D stylization results compared to other methods. Image style transfer approaches produce more noticeable inconsistency artifacts than the other two, since each frame is stylized independently. Video-based approaches perform better than the image-based approaches since they take into short-term consistency. However, MCCNet [Deng et al. 2021] still produces noticeable artifacts when two frames are far apart, and ReReVST [Wang et al. 2020a] does not capture the reference style as well.

For 3D-based approaches, we observe that StyleScene [Huang et al. 2021], Chiang et al. [2022] and our approach generate view-consistent results since all methods aim to stylize a holistic 3D scene. (Note that StyleScene results from the authors' model are at size 538×274.) However, StyleScene's results do not capture the reference style image as well as ours, as also shown in the user study in Section 4.2.1 and Figure 3. Similarly, Chiang et al. [2022]'results fail to capture the reference style well, such as the overall colour schemes or the fine-grained stippling details (Figure 3 left). This can be mostly explained by the fact that both of these methods stylize

only scenes' *appearance* instead of both geometry and appearance (see ablation study in Section 4.2.3). Additionally, Chiang et al. [2022] only trained with small patches of size 81×67 out of 1008×550 images, which has been shown to produce results that lack global structural coherence [Huang and Belongie 2017; Texler et al. 2019] or diversity [Wang et al. 2021]. Finally, it is non-trivial to extend StyleScene to dynamic scenes, whereas our method can be directly applied to stylize 4D dynamic avatars (see Figure 6).

Figures 1, 4, 5 and 6 show additional qualitative results of our method on different scene types. Although we choose to use the original stylization approach by Gatys et al. [2016] in this work, our scene stylization framework is not restricted to a particular stylization technique. For example, to stylize the dynamic avatar in Figure 1, we use ArcaneGAN [Spirin 2021], which is built upon a memory-intensive StyleGAN model [Karras et al. 2019]. This makes it challenging when naively combining ArcaneGAN with NeRF to perform scene stylization. However, thanks to our alternating training approach, we can easily adopt this model to stylize a dynamic 4D avatar and produce results at high resolution (512×512).

4.2 Quantitative results

4.2.1 User study. We conduct a user study to compare the user preference between our proposed and alternative approaches. In particular, we want to measure users' preferences in two aspects, split into two tests: (1) which method produces more consistent results across different views (e.g., less flickering), and (2) which method matches the style of a given reference style image better. For each question, we ask the participant to compare two videos of the same scene and style, one generated by our method and the other by one alternative method. To generate the videos for the study, we stylize 5 scenes: *Fern*, *Truck*, *Train*, *Playground* and *M60*. We collect answers from 35 participants for both questions. As shown in Figure 7, our method (coloured in grey) performs better than other approaches on both stylization quality and multi-view consistency.

4.2.2 Cross-view consistency. One of the main advantages of using implicit scene representation for stylization is cross-view consistency. To test the performance of our approach, we adopt a similar strategy to Lai et al. [2018] to measure the consistency between different novel views of the stylized 3D scene. In particular, we create testing videos where each frame is a rendered image at a novel view of our stylized scene. We then compute the optical flow and the occlusion mask O using two ground truth views I_i^{real} and $I_{i+\delta}^{real}$ rendered from a pre-trained NeRF scene. Note that unlike Lai et al. [2018] who use FlowNet2 [Ilg et al. 2017], we use RAFT [Teed and Deng 2021], a state-of-the-art method to predict optical flow between two views. Secondly, we warp a stylized view I_i^s to obtain a new view $\hat{I}_{i+\delta}^s$ using the optical flow. Finally, we compute the error between the novel view obtained from our stylized scene $I_{i+\delta}^s$, and the previously computed $\hat{I}_{i+\delta}^s$ using:

$$E_{consistency}(I_{i+\delta}^s, \hat{I}_{i+\delta}^s) = \frac{1}{|O'|} \left\| \dot{I}_{i+\delta}^s - \hat{I}_{i+\delta}^s \right\|_2^2 \qquad (6)$$

where $|O'|$ denotes the number of non-occluded pixels calculated using O. Following Huang et al. [2021] and Chiang et al. [2022], we

Fig. 3. **Qualitative comparisons**. We encourage readers to look at the supplementary material to compare the consistency between different methods. We show stylized results from two frames that are far apart (t^{th} and $(t+8)^{th}$ frame). Cross-view inconsistencies are highlighted in blue boxes.

Fig. 4.   SNeRF's stylization results on 360°scenes using NeRF++. Here we show our results from different views.
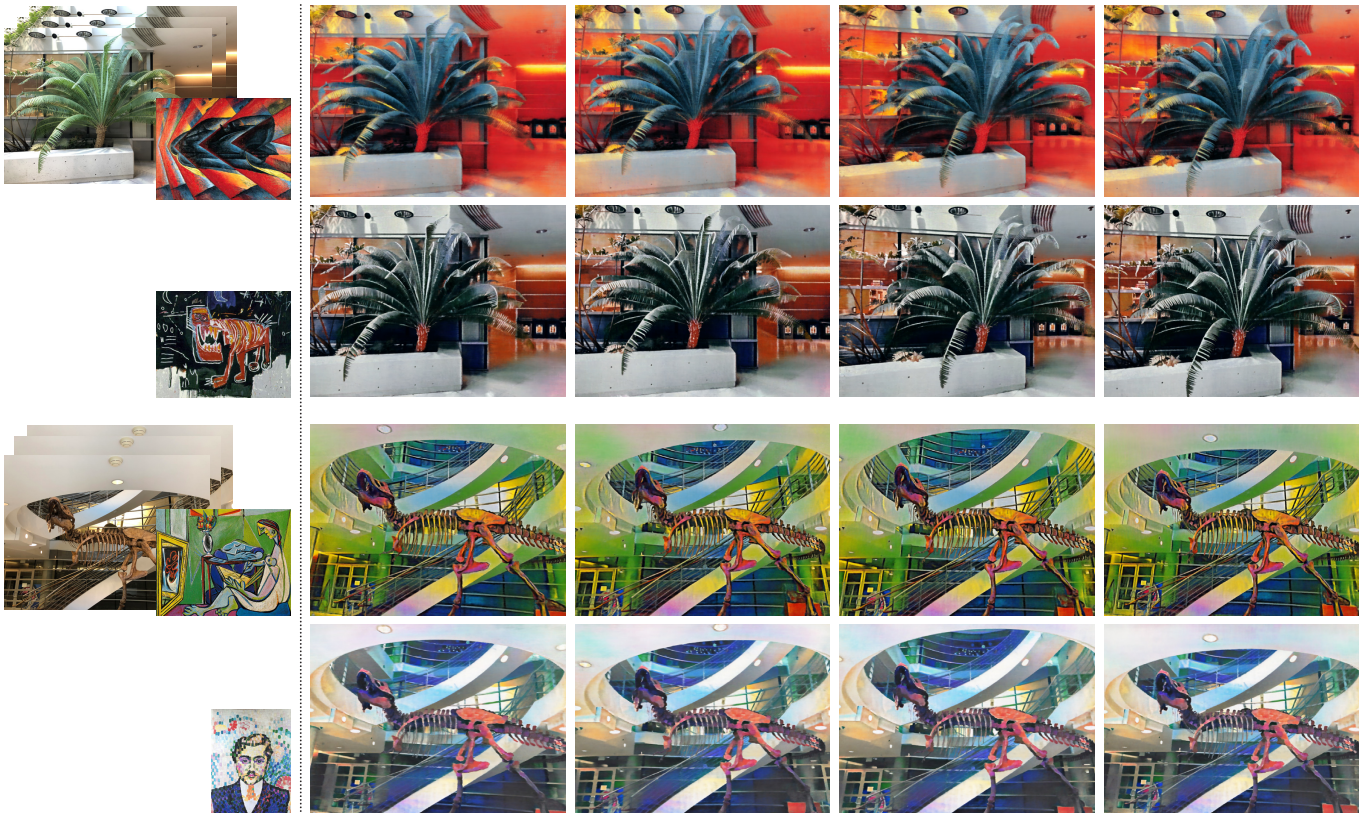

Fig. 5.   SNeRF's stylization results on indoor scenes using NeRF. Here we show our stylization results from different views.

Fig. 6. SNeRF's stylization results for a dynamic avatar. Using neural implicit representations, which are compact and flexible, allows us to seamlessly extend our method to stylize this 4D dynamic avatar. Our stylized avatar can generate results that are consistent across different views and expressions.
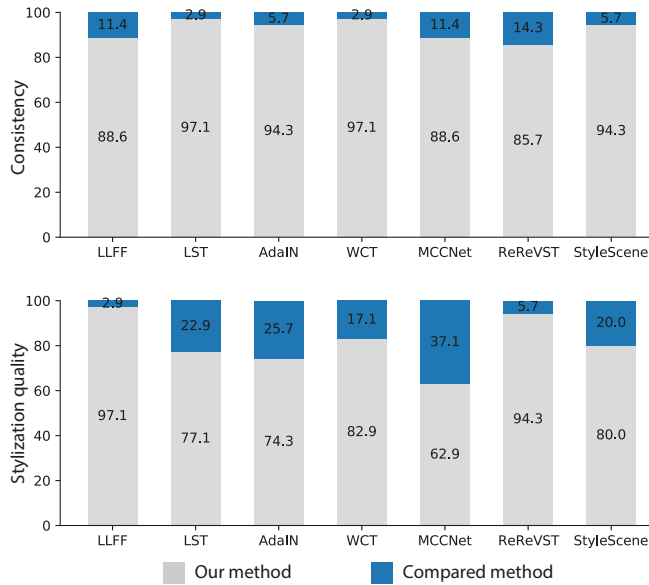

Fig. 7. User preference study. We present two videos of novel view synthesis results, one generated by our method (grey) and one by another approach (blue - each column corresponds to one approach we are comparing against). We ask the participant to select the one that (a) shows less flickering artifacts and (b) matches the reference style image better.

measure both *short-range* and *long-range* consistency between different testing video frames. Specifically, we compute the error between i^{th} and $(i + 1)^{th}$ video frames to measure *short-range* consistency, and between i^{th} and $(i + 7)^{th}$ frames for *long-range* consistency.

We show our results for short and long-range consistency measurement in Table 1 and 2 respectively. We compare our method with image-based approaches (AdaIN, WCT and LST), as well as video-based approaches (ReReVST and MCCNet), and report the average errors of 12 diverse styles. Unfortunately, we could not match the camera path and resolution for the results of StyleScene [Huang et al. 2021], and thus do not include this work in this comparison. In general, the image stylization alternative methods produce worse results than video-based and 3D-based methods (ours). We observe

that ReReVST produces competitive results with our method. However, as shown in Figure 3, and user study in Section 4.2.1, ReReVST does not capture the reference style well. As shown in the user

Table 1. Qualitative comparisons on short-range consistency. We compute the consistency score (the lower the better) between two nearby stylized novel views. The best result is in bold and the second best is underscored.

Methods	Truck	Playground	M60	Train
AdaIN	0.043	0.044	0.054	0.039
WCT	0.064	0.063	0.084	0.056
LST	0.027	0.026	0.032	0.024
ReReVST	0.010	0.009	**0.010**	0.015
MCCNet	0.025	0.025	0.028	0.021
SNeRF (Ours)	**0.009**	**0.004**	0.012	**0.008**

Table 2. Qualitative comparisons on long-range consistency. We compute the consistency score (the lower the better) between two far-away stylized novel views. The best result is in bold and the second best is underscored.

Methods	Truck	Playground	M60	Train
AdaIN	0.059	0.060	0.075	0.062
WCT	0.084	0.087	0.110	0.082
LST	0.037	0.032	0.042	0.036
ReReVST	**0.015**	0.015	**0.016**	0.024
MCCNet	0.035	0.030	0.039	0.034
SNeRF (Ours)	0.026	**0.010**	0.032	**0.016**

studies and consistency measurements, SNeRF can stylize 3D scenes to generate novel views faithful to both the reference style and the original scene content while maintaining cross-view consistency.

4.2.3 Freezing geometry. Recent 3D scene stylization approaches focus only on stylizing the appearance instead of both the underlying geometry and appearance [Chiang et al. 2022; Huang et al.
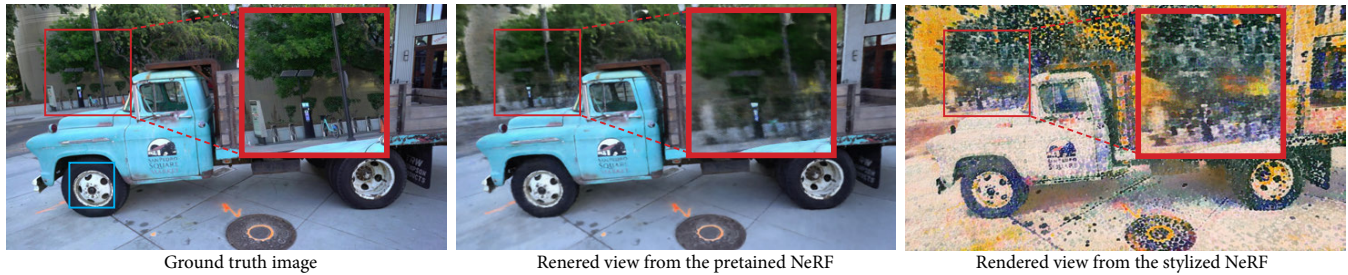
Fig. 8. The quality of our stylized results partly depends on the quality of the underlying implicit scene function. If the scene function fails to capture sharp details (shown in the red box), the stylized results will be blurry.

Fig. 9. The effects of only updating the appearance of NeRF. Stylizing both geometry and appearance leads to results with sharper details and closer to the reference style, especially for more abstract reference style images.
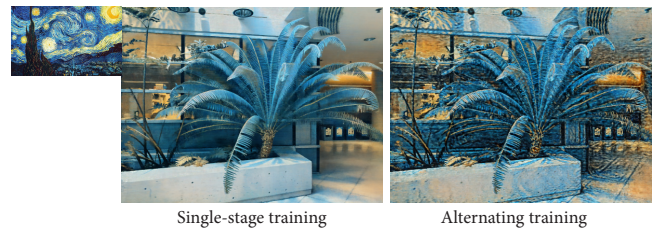

Fig. 10. Alternating between training NeRF and stylization leads to better results than a single-stage training using individually stylized images.

2021]. However, Liu et al. [2021] and Kim et al. [2020] show that geometry is also a component of style, and thus stylizing both style and geometry leads to stylized results that are closer to the target style. Therefore, our method stylizes both the appearance and geometry (represented as density) of the scene functions. Figure 9 shows that this produces stylized results that are closer to the target style, especially when the style is more abstract or contains lots of fine-grained details. Meanwhile, when we only stylize the appearance (by keeping the weights of NeRF's shared and opacity branch fixed, and only updating the weights of the RGB branch), the results only capture the style's color scheme.

4.2.4 Alternating training scheme. In addition to addressing the memory limitations, our alternating training framework can also produce stylization results that capture the reference style better. The 3D scene stylization process comprises two main steps: image stylization, which allows us to perform stylization using a reference image but does not guarantee multi-view consistency, and scene stylization, which modulates the scene to match the set of stylized views and maintain consistency. In our method, we alternate between these two steps for a few iterations, where one iteration comprises stylization and NeRF training.

We show that naively training a scene function using a set of inconsistent stylized images leads to cross-view consistent results, but fails to capture the target style, similar to the novel view synthesis results by LLFF. This is the equivalent of performing only one iteration of our approach. However, as shown in Figure 10, if we repeat the same process of stylization and NeRF training for more iterations, we get better results that are multi-view consistent and capture fine-grained details of the reference style.

5 DISCUSSION

In this paper, we have shown both qualitatively and quantitatively the advantages of SNeRF in terms of generating multi-view consistent results, compared to image and video style transfer methods [Deng et al. 2021; Huang and Belongie 2017; Li et al. 2019, 2017; Wang et al. 2020a]. We also show that our method generates better stylization results than other 3D-based approaches [Chiang et al. 2022; Huang et al. 2021], which can be mostly attributed to our method's flexibility in stylizing both the 3D scene geometry and appearance. Unlike these existing methods that focus on static scenes, our method can also stylize dynamic content, such as 4D avatars.

We observe that the quality of our stylized results partly depends on the quality of the scene function trained with RGB images, as also observed by Chiang et al. [2022]. For example, in Figure 8, the underlying NeRF model fails to capture sharp details of the vegetation in the background, compared to the ground truth RGB image. This eventually leads to blurry stylized results.

Our method adopts NeRF which is more time-consuming and computationally demanding than point clouds. Depending on the resolution of the scene, on a single Nvidia V100 GPU, training (including training the original NeRF models) can take 3-5 days and rendering an image at size 1008×548 can take 55 seconds. We believe that our alternating stylization framework will enable quick adoption of fast emerging advances in research on NeRF to improve quality [Barron et al. 2021, 2022] and speed [Hedman et al. 2021a; Müller et al. 2022; Neff et al. 2021], as well as style transfer methods to improve the fidelity and variety of stylization results [Gal et al. 2021; Texler et al. 2019]. Note that, even though it takes a while to train a stylized NeRF using our method, once trained, they can be

"baked" [Hedman et al. 2021b] for real-time rendering in AR, VR and MR applications.

Apart from RGB images, future work can explore using additional segmentation or depth maps (queried from NeRF models) for stylization [Liu et al. 2017; Wang et al. 2020b]. Secondly, we currently stylize each scene independently, and cannot apply an arbitrary style to each scene without restarting the optimization process. Therefore, it will be an interesting direction to combine our framework with recent work on arbitrary style transfer.

6 CONCLUSION

In this work, we present a method for 3D scene stylization using implicit neural representations (NeRF). This provides a strong inductive bias to produce stylized multi-view consistent results that also match a target style image well. Additionally, our alternating stylization method enables us to make full use of our hardware memory capability to stylize both static and dynamic 3D scenes, allowing us to both generate images at higher resolution and adopt more expressive image style transfer methods. As NeRF increasingly attracts more research in improving generalisation, quality, and speed in both training and test time, we believe that using implicit scene representations for 3D scene stylization will open up a wide range of exciting applications for AR, VR and MR.

REFERENCES

Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. 2020. Neural point-based graphics. In *Proceedings of the European Conference on Computer Vision*. 696–712.

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 5855–5864.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022).

Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive Light Field Video with a Layered Mesh Representation. *ACM Trans. Graph.* 39, 4 (July 2020), 86:1–15.

Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. 2001. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. 425–432.

Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura. 2020. PSNet: A Style Transfer Network for Point Cloud Stylization on Geometry and Color. In *The IEEE Winter Conference on Applications of Computer Vision*.

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth Synthesis and Local Warps for Plausible Image-Based Navigation. *ACM Trans. Graph.* 32, 3, Article 30 (jul 2013).

Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. 2017. Coherent Online Video Style Transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1114–1123.

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. 2018. Stereoscopic Neural Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xinghao Chen, Yiman Zhang, Yunhe Wang, Han Shu, Chunjing Xu, and Chang Xu. 2020. Optical Flow Distillation: Towards Efficient and Stable Video Style Transfer. In *Proceedings of the European Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 12351)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 614–630.

Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. 2022. Stylizing 3D Scene via Implicit Representation and HyperNetwork. (January 2022), 1475–1484.

Frank Dellaert and Lin Yen-Chen. 2021. Neural Volume Rendering: NeRF And Beyond. arXiv:2101.05204 [cs.CV]

Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. 2021. Arbitrary Video Style Transfer via Multi-Channel Correlation.

Proceedings of the AAAI Conference on Artificial Intelligence 35, 2 (May 2021), 1210–1217.

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 Trans. Graph.* 35, 4, Article 92 (jul 2016), 11 pages.

John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5515–5524.

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2021. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8649–8658.

Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. 2021. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. arXiv:2108.00946 [cs.CV]

Chang Gao, Derun Gu, Fangjun Zhang, and Yizhou Yu. 2018. ReCoNet: Real-time Coherent Video Style Transfer Network. In *Proceedings of the 18th Asian Conference on Computer Vision*, Vol. abs/1807.01197.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2414–2423.

Xinyu Gong, Haozhi Huang, Lin Ma, Fumin Shen, Wei Liu, and Tong Zhang. 2018. Neural Stereoscopic Image Style Transfer. In *Proceedings of the European Conference on Computer Vision*.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. 1996. The Lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. 43–54.

Filip Hauptfleisch, Ondřej Texler, Aneta Texler, Jaroslav Křivánek, and Daniel Sýkora. 2020. StyleProp: Real-time Example-based Stylization of 3D Models. *Computer Graphics Forum* 39, 7 (2020), 575–586.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. 2018. Deep Blending for Free-Viewpoint Image-Based Rendering. *ACM Trans. Graph.* 37, 6, Article 257 (dec 2018).

Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021a. Baking Neural Radiance Fields for Real-Time View Synthesis. (October 2021), 5875–5884.

Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021b. Baking Neural Radiance Fields for Real-Time View Synthesis. (October 2021), 5875–5884.

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*. 327–340.

Lukas Höllein, Justin Johnson, and Matthias Nießner. 2021. StyleMesh: Style Transfer for Indoor 3D Scene Reconstructions. *CoRR* abs/2112.01530 (2021). arXiv:2112.01530

Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. 2017. Real-Time Neural Style Transfer for Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7044–7052.

Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. 2021. Learning To Stylize Novel Views. (October 2021), 13869–13878.

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision*.

E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

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, Article 107 (2019).

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*.

Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. 2016. Learning-Based View Synthesis for Light Field Cameras. *ACM Trans. Graph.* 35, 6, Article 193 (nov 2016).

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Sunnie S. Y. Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. 2020. Deformable Style Transfer. In *Proceedings of the European Conference on Computer Vision*.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Trans. Graph.* 36, 4, Article 78 (jul 2017), 13 pages.

Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. 2021. Point-Based Neural Rendering with Per-View Optimization. *Computer Graphics Forum*

(Proceedings of the Eurographics Symposium on Rendering) 40, 4 (June 2021).

Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning Blind Video Temporal Consistency. In *Proceedings of the European Conference on Computer Vision.* 179—-195.

Marc Levoy and Pat Hanrahan. 1996. Light Field Rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques.* 31–42.

Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. 2019. Learning Linear Transformations for Fast Image and Video Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Universal Style Transfer via Feature Transforms. In *Advances in Neural Information Processing Systems.*

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2021. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 6498–6508.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual Attribute Transfer through Deep Image Analogy. *ACM Trans. Graph.* 36, 4, Article 120 (jul 2017), 15 pages.

Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L. Rosin. 2017. Depth-Aware Neural Style Transfer. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (Los Angeles, California) *(NPAR '17).* Association for Computing Machinery, New York, NY, USA, Article 4, 10 pages.

Xiao-Chang Liu, Yong-Liang Yang, and Peter Hall. 2021. Learning to Warp for Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3701–3710.

Chongyang Ma, Haibin Huang, Alla Sheffer, Evangelos Kalogerakis, and Rui Wang. 2014. Analogy-Driven 3D Style Transfer. *Computer Graphics Forum* 33, 2 (2014), 175–184.

Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 7210–7219.

Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019a. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 29:1–29:14.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019b. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* (2019).

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of the European Conference on Computer Vision.* 405–421.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *arXiv:2201.05989* (Jan. 2022).

Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. 2021. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum* 40, 4 (2021).

Chuong H. Nguyen, Tobias Ritschel, Karol Myszkowski, Elmar Eisemann, and Hans-Peter Seidel. 2012. 3D Material Style Transfer. *Computer Graphics Forum (Proc. EUROGRAPHICS 2012)* 2, 31 (2012).

Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–11.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). 8748–8763.

Mattia Segu, Margarita Grinvald, Roland Siegwart, and Federico Tombari. 2020. 3DSNet: Unsupervised Shape-to-Shape 3D Style Transfer. *arXiv preprint arXiv:2011.13388* (2020).

Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. 2019. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2437–2446.

Alex Spirin. 2021. *ArcaneGAN.* https://github.com/Sxela/ArcaneGAN

Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition. 175–184.

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. 2020. Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

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

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

Zachary Teed and Jia Deng. 2021. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow (Extended Abstract). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21,* Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4839–4843.

Ondřej Texler, Jakub Fišer, Michal Lukáč, Jingwan Lu, Eli Shechtman, and Daniel Sýkora. 2019. Enhancing Neural Style Transfer using Patch-Based Synthesis. In *Proceedings of the 8th ACM/EG Expressive Symposium.* 43–50.

Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. 2020. Interactive Video Stylization Using Few-Shot Patch-Based Training. *ACM Transactions on Graphics* 39, 4 (2020), 73.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–12.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR* abs/1607.08022 (2016). arXiv:1607.08022

Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu. 2020a. Consistent Video Style Transfer via Relaxation and Regularization. *IEEE Trans. Image Process.* (2020).

Zhizhong Wang, Lei Zhao, Haibo Chen, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. 2021. Diversified Patch-based Style Transfer with Shifted Style Normalization. *CoRR* abs/2101.06381 (2021). arXiv:2101.06381

Zhizhong Wang, Lei Zhao, Sihuan Lin, Qihang Mo, Huiming Zhang, Wei Xing, and Dongming Lu. 2020b. GLStyleNet: exquisite style transfer combining global and local pyramid features. *IET Computer Vision* 14, 8 (2020), 575–586.

Xide Xia, Tianfan Xue, Wei-sheng Lai, Zheng Sun, Abby Chang, Brian Kulis, and Jiawen Chen. 2021. Real-time Localized Photorealistic Video Style Transfer. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV).* 1088–1097.

Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 2021. 3DStyleNet: Creating 3D Shapes With Geometric and Texture Style Variations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 12456–12465.

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv:2010.07492* (2020).

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. 2004. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)* 23, 3 (2004), 600–608.