StyleTRF: Stylizing Tensorial Radiance Fields*

Rahul Goel†
rahul.goel@research.iiit.ac.in
CVIT, KCIS, IIIT Hyderabad
India

Sirikonda Dhawal†
dhawal.sirikonda@research.iiit.ac.in
CVIT, KCIS, IIIT Hyderabad
India

Saurabh Saini
saurabh.saini@research.iiit.ac.in
CVIT, KCIS, IIIT Hyderabad
India

P. J. Narayanan
pjn@iiit.ac.in
CVIT, KCIS, IIIT Hyderabad
India

Figure 1: *Stylization:* We show results of stylization using our technique presented in Sec. 3. We stylize the Play-Ground scene using two different styles *mosaic* and *mudbath*. Our Pipeline adapts style in a nominal time of 40 sec on top of a pre-optimized TensoRF scene representation. Once the style is adapted in accordance, stylized novel views can be generated with traditional volumetric rendering techniques.

ABSTRACT

Stylized view generation of scenes captured casually using a camera has received much attention recently. The geometry and appearance of the scene are typically captured as neural point sets or neural radiance fields in the previous work. An image stylization method is used to stylize the captured appearance by training its network jointly or iteratively with the structure capture network. The state-of-the-art SNeRF [29] method trains the NeRF and stylization network in an alternating manner. These methods have high training time and require joint optimization. In this work, we present StyleTRF, a compact, quick-to-optimize strategy for stylized view generation using TensoRF [6]. The appearance part is fine-tuned using sparse stylized priors of a few views rendered using the TensoRF representation for a few iterations. Our method thus effectively decouples style-adaption from view capture and is much faster than the previous methods. We show state-of-the-art results on several scenes used for this purpose.

CCS CONCEPTS

• **Computing methodologies** → **Appearance and texture representations**; Computational photography; **Volumetric models**; *Non-photorealistic rendering*.

KEYWORDS

NeRF, Content Stylization, Multi-view consistency, Fine-tuning, Fast Adaptation

ACM Reference Format:
Rahul Goel, Sirikonda Dhawal, Saurabh Saini, and P. J. Narayanan. 2022. StyleTRF: Stylizing Tensorial Radiance Fields. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'22), December 8–10, 2022, Gandhinagar, India,* Soma Biswas, Shanmuganathan Raman, and Amit K Roy-Chowdhury (Eds.). ACM, New York, NY, USA, Article 43, 9 pages. https://doi.org/10.1145/3571600.3571643

1 INTRODUCTION

Stylizing a content image based on a reference style image has been of interest to the community lately. With the development of 3D visual devices, the demand for 3D content generation has grown. Stylizing entire 3D scenes has applications in the world of augmented reality (AR) and virtual reality (VR). With style transfer on 3D scenes, one can witness an entire world through a painter's eyes.

*Project Page
†Both authors contributed equally to this research.

ICVGIP'22, December 8–10, 2022, Gandhinagar, India

ACM ISBN 978-1-4503-9822-0/22/12.
https://doi.org/10.1145/3571600.3571643

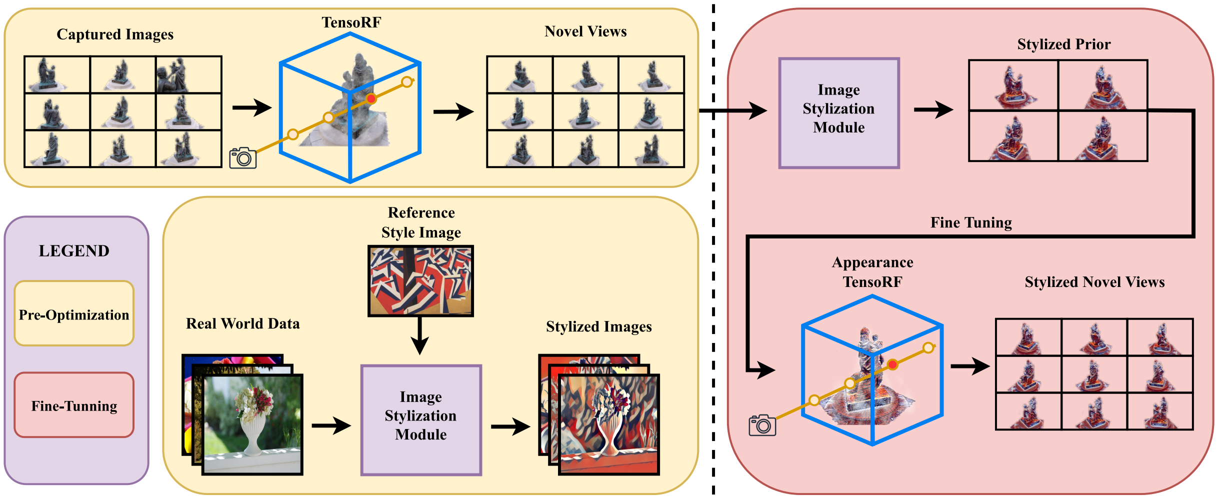


Figure 2: *System overview*: The pipeline diagram presents an overview of the strategy employed by our work. We first optimize Tensorial Radiance Fields for representation of the scene as proposed by [6]. Concurrently we optimize stylization module utilizing Johnson et al. [20] method on COCO14-dataset[24]. The stylization module is then used to stylize a sparse set of novel views generated by the pre-optimized TensoRF. These stylized views act as sparse style-priors and are used to fine-tune the appearance of the previously optimized scene representation. It is to be noted that we freeze the density terms of the TensoRF and only alter the appearance vectors which retains geometric while adapting the novel style.

There have been several efforts to stylize image content [10, 18, 20, 23, 33, 35]. Stylizing videos on the other hand is a harder problem as it needs temporal consistency along with stylizing across the frames of the video [16, 25, 31, 38]. Combining stylization with new view generation takes the game one step further. We concentrate on stylized novel view generation in this paper.

Radiance Fields have become the dominant method of capturing a scene using a few images and then generating its views from other positions. NeRF [28] and its successors exploit the neural representations to encode the whole scene and render novel views at test time. They use a small MLP-based architecture to represent the radiance field (typically using 8 layers of 256 neurons), which has a small memory footprint of 5MB. They, however, need very high training time, ranging between 16hrs to 1 day per scene. Plenoxels [9] leverage the traditional voxel space to optimize for the feature vectors associated per voxel and rely on standard tri-linear interpolation to regress the radiance field vectors for the continuous volumetric space. The voxel-based approach benefits from the adaptive upsampling and pruning techniques, thereby reducing the optimization times to about 15-minutes. Plenoxels has a large memory footprint, often running into GBs. The recently introduced TensoRF method [6] uses the Tensor decomposition based on BTD and provides a way to reduce both memory and training times simultaneously. TensoRF can optimize a scene in 10-20 minutes while maintaining a memory footprint of just 5-10MB.

Recent stylized new view generation methods use NeRF to represent the scene. They jointly or iteratively optimize NeRF with a stylization technique while rendering stylized new views [7, 19, 29]. The latest method SNeRF proposed by Nguyen-Phuoc et al. [29] uses an iterative training strategy to optimize for each style using Gatys et al. [10]. Adapting to a new style is thus a heavy process, taking days of processing.

In this work, we build on and modify the ideas of SNeRF [29] and present a simple yet powerful method for stylized new view generation. Our method nearly decouples the scene capture step from the stylization step. Each can be performed in respective preprocessing step. The joint optimization is limited to a few iterations of appearance fine-tuning that takes about 40-50 seconds to adapt to a new style. We use TensoRF [6] to represent the geometry and appearance of the scene, which comes with low optimization time and a small memory footprint. We use the method by Johnson et al. [20] for stylization. It can be trained for a particular style image (typically in under 20 minutes) ahead of time and can produce stylized images at a rate of 30 frames per sec. We generate a few novel views using the learned TensoRF representation and stylize them using Johnson's method. We then fine-tune the TensoRF's appearance branch while freezing the density components for a few iterations to obtain a stylized scene representation. Consistent, stylized novel views of the scene can be generated using this fine-tuned TensoRF representation.

The main contribution of this work is using fast and light TensoRF representation for stylized view generation and devising a fast style-adaptation technique. We do this by fine-tuning the TensoRF appearance for a small number of iterations, using as prior a sparse set of images rendered from the optimized TensoRF and stylized using Johnson et al. [20]. This effectively decouples style-adaption from view capture and makes it much faster than methods like SNeRF.

2 RELATED WORK

Stylizing Images: Stylizing images has been a well-studied problem in the vision community for a long time. The method proposed by Gatys et al. [10] optimizes white noise to match the content of one image while transferring the style from the other. Johnson et al. [20] proposed to use simple feed-forward architecture, which produces stylized content in real-time. While being quick at producing results, Johnson et al. [20] require a separate network for each style. Several works like [18, 23, 33] have addressed this issue of style-dependent training. More recently, Svoboda et al. [35] have been able to decompose an image into its style and content codes. While these works have provided a strong basis for stylizing 2D content, most of the aforementioned works do not faithfully stylize temporal data, let alone stylizing 3D content.

Temporal Stylization of Videos: The stylization of temporal-data-like videos necessitate temporal-consistent stylization. Though 2D stylization techniques provide good stylization per frame, when stitched together, they often result in flickering. This flicker is caused due to inconsistent temporal stylization. To address these issues, various methods have tried incorporating temporal-consistency losses across frames, namely Ruder et al. [31]. Further works like Huang et al. [16] concentrate on the real-time generation of stylization in addition to temporal-consistent stylization. Recent work in this line by Wang et al. [38] has relaxed the objective function to be optimized and derived a new regularization strategy. This allowed them to stylize video content for any arbitrary given style without any re-training or fine-tuning. Though these methods provide temporally-consistent stylization, they can not produce novel stylized views. This requires scene understanding at the geometric and as well as at the appearance levels. Consequently, temporal consistency methods proposed by video stylization works cannot be extended to novel view stylization.

Stylizing Geometric Appearance: As discussed earlier, obtaining stylized novel views require an understanding of the geometric content of the scene. Methods like Cao et al. [5] have proposed stylizing geometry represented as 3D-point-cloud referenced on another point-cloud or image. However, stylizing point-cloud do not fully address the problem of novel view stylization as they are often accompanied by undesirable noise and missing surface information. These noise points and missing geometries can only be filled up using neural rendering-based frames works. Works like NPBG [1] used U-Net-based architecture to fill the missing gaps in geometry and alleviate the noisy data. Moving in this direction Huang et al. [17] introduced consistent 3D point cloud stylization through a learned linear transformation matrix to change the appearance. They build up on NPBG [1] to account for the sparsity and noisy nature of point clouds for the rendering of stylized novel views. Though the method proposed by Huang et al. [17] produces promising stylization of novel views, the geometric representation and appearance captured by the underlying U-Net-based pipelines are not accurate.

Novel View Synthesis(NVS) using Implicit Neural Representation: The Novel View Synthesis(NVS) has been a challenging field that has been tackled for decades with different approaches like Lumigraph [13], Multi-Plane Imaging [37], and Light Field Fusion [27]. Recently the field of NVS has taken a large leap forward due to the introduction of NeRF[28]. NeRF(Neural Radiance Fields) proposed by Mildenhall et al. [28], learns the radiance fields using simple MLPs coupled with positionally encoded input features to regress radiance at novel viewpoints, given a sparse set of posed images. Despite its great representational power, it is accompanied by limitations like slow training and rendering times, non-editable appearance, and lighting. These limitations have paved the path for various extensions ranging from speeding up its rendering process [15, 30], edit-ability of material [3, 4, 40], relighting [4, 34] and improving implicit geometry [21]. Recently HDR based NeRFs have also garnered great interest [11, 26]. Unlike these methods, we concentrate on stylizing Radiance fields with reference style images.

Novel View Synthesis(NVS) using voxel grids: Apart from Implicit Neural representation of radiance fields, recently, PlenOxels [9] proposed the use of voxel grid with density and appearance associated in the form SH-vectors stored at every voxel of the grid to represent the underlying radiance field. They utilized custom CUDA implementation to achieve a training time of around 20 minutes while obtaining similar results as NeRF. However, their approach requires per-scene storage of around 1GB. Addressing the issue of memory footprint and requirement of custom CUDA kernels, TensoRF [6] has proposed a new VM-(Vector Matrix) Decomposition (a special case of Block Term Decomposition) [8, 12]. The usage of VM-decomposition alleviated the issue of a large memory footprint while maintaining similar scene optimization times as PlenOxels.

Stylizing Radiance Fields Recently, Chiang et al. [7] extended NeRF to generate stylized novel views. Similar to NeRF, they rely on simple MLPs to regress radiance fields, compositing density and radiance. To achieve stylization, they split the representation (1) volumetric density MLP and (2) appearance MLP. They use a hypernetwork [14] trained on the style to predict the weights of the appearance MLP which introduces style information into the appearance of the scene. The losses involved in their optimization framework are memory intensive. To circumvent this issue they optimize the losses in a patch-based manner. Due to the usage of patch-based loss, their method suffers from inconsistencies across the patch borders in a single image frame, which is undesirable. Along with patch artifacts, due to different MLP splits(density and appearance), Chiang et al. [7] require large training and rendering times.

Recent work StylizedNeRF [19] approaches the problem of stylizing radiance fields by mutually optimizing 2D-3D consistency. For incorporating style, they replace the appearance MLP of NeRF with a style module. They also introduce consistency and mimic losses to train the style module and a 2D stylization decoder simultaneously. However, their style transfer is not accurate, and their approach involves nearly a day-long training.

SNeRF [29] devises a new training strategy by incrementally stylizing novel views. They first train a NeRF model on real-world data and generate novel views using it. Using the approach by Gatys et al. [10], they generate a stylized version of these novel views based on a reference style. Then the NeRF model is re-trained NeRF on the so-obtained stylized views. Iteratively repeating the previous two steps stylizes a Neural radiance field representation to incorporate style information. Once the scene is stylized novel views can be generated using traditional volumetric rendering techniques. However, this learning of radiance fields involves optimization of the Neural implicit representation which comes with a heavy training cost of 4-5 days.

We take inspiration from SNeRF and train a radiance field representation on stylized images themselves. However, we realize the limitations that Gatys's method introduces when generalizing the 2D images to a 3D domain and propose the usage of a different stylization module.

3 METHOD

Given a set of posed images of a scene and a reference style image, our aim is to render stylized novel views of the scene, which are consistent in appearance and geometry across different frames. We achieve this by fine-tuning the appearance of a TensoRF [6] using stylized images.

3.1 Preprocessing Stylization Module

In our method, we use the stylization method presented by Johnson et al. [20], which produces stylized content in real time. The approach requires a per-style training of a CNN, for which it utilizes COCO2014 dataset [24]. The per-style optimization takes an approximate time of 20-minutes and at the inference, it produces 30 images/sec. Since training per-style takes only 20 minutes, we can simultaneously train multiple Johnson et al. [20] models for each desirable style independently and infer based on the preference.

We choose Johnson et al. [20] as our underlying stylization module over Gatys et al. [10], unlike others [29] because it gives stable stylization for two closeby viewpoints. In the case of Gatys et al. [10], stylizing two closeby viewpoints that share a large amount of image content usually results in drastically different stylization as shown in Fig. 3. This is because the image developed using Gatys et al. [10] start with white-noise and try to converge the distribution of *reference-style* and *content* images. This results in unstable stylization across nearby viewpoints as the optimization depends on various factors such as the learning rate, initialization, and type of optimizer. One of the reasons for the slow multi-iter stylization in SNeRF was to alleviate the inconsistencies caused by Gatys et al. [10] based stylization.

On the other hand, once Johnson et al. [20] is trained, the output image is deterministic with respect to the input image. Furthermore, two close-by camera viewpoints share a large amount of spatial information. Since CNN's are spatially invariant in the local region, our image transformation network gives fairly stable stylization across closeby viewpoints Fig. 3.

Though Johnson et al. [20] provide stable stylization across adjacent views, it is *not temporally consistent.* Hence it is to be noted that, we do not rely on temporally-consistent stylization while only relying on the nominally-stable stylization of nearby viewpoints. The per-style training of our Stylization Module is depicted in Fig. 2.

3.2 Scene Representation using TensoRF

In order to generate a novel view, it is necessary to have a geometric representation and appearance understanding. In order to address this issue, we use Radiance fields($\mathcal{L}$). The radiance fields [41] ($\mathcal{L}$) is given by:

$$\mathcal{L} : R^3 \times S^2 \rightarrow R^3 \tag{1}$$

where R^3 on the left is the scene's world space; S^2 is the sphere of directions about each point, and the R^3 on the right is radiance at the point. Radiance fields in a way encode geometry and radiance in their mapped representation. Though there exist various recent adaptions of the mapping function presented in Eq.(1) like NeRF[28], PlenOxel[9], we chose TensoRF [6] which is a compact, accurate, and fast-to-optimize representation of Radiance Fields. We use a specifically *VM-48* variant of TensoRF which optimizes a scene within a timeframe of 15-20 minutes while maintaining a memory footprint of 10-15MB.

The TensoRF representation utilizes a Tensor decomposition known as VM-decomposition which in itself is a special case of BT-decomposition. This reduces the voxel grid memory by order of $O(n)$. This scene optimization can be done independently for every scene irrespective of style. In a later phase, we alter the appearance to adapt to the reference style in a short period of 40-50 seconds.

Similar to TensoRF [6], we utilize $L1$ norm loss and total variation (TV) loss (Eq. 2) for regularization. This helps our process to avoid getting stuck in local minima and prevents overfitting. For scenes with less number of captured images, TV loss is a better choice to obtain good results. The equation for TV loss is given by:

$$\mathcal{L}_{TV} = \frac{1}{N} \sum \left(\sqrt{\Delta^2 T_\sigma} + 0.1\sqrt{\Delta^2 T_C} \right) \tag{2}$$

Here, Δ^2 is the squared difference between the neighboring values in our tensors, N is the total number of parameters across our TensoRF representation T. T_σ represents the density value and T_C represent the appearance value in the TensoRF representation. They are weighted in the ratio of $10 : 1$ respectively. More details about TV Loss can be found in the work by Chen et al. [6].

3.3 Stylizing TensoRF representation

3.3.1 *Novel View stylization.*

Upon optimizing the radiance fields which encode the geometry and radiance of the scene as discussed in the Sec.3.2, we render a sparse set of $20 - 30$ novel views in a simple trajectory (spiral). We stylize these novel renders using the pre-trained Stylization Module discussed in 3.1. Fig. 2 (top-right) shows the generation stylization of these renders utilizing the Stylization Module.

3.3.2 *Stylizing Appearance of TensoRF.*

We utilize the sparse set of stylized novel views generated using the per-style optimized Johnson et al. [20] module and optimize the appearance vectors of the TensoRF. During the process of optimization, we ensure that the density terms are frozen, and only the appearance is altered. We explicitly chose to freeze density as we have observed that stylizing looks pleasing and free from artifacts when density is kept frozen. This fine-tuning only takes a nominal time of downwards of 40 secs. Once the fine-tuning is done, we obtain a geometric scene represented as Tensorial Radiance fields, which can be used to render *stylized novel views* with consistent appearance across the viewpoints. The rendering of each image having a resolution of 800×800 takes an approximate time of 4-5 seconds. Fig. 2 (bottom-right) shows the appearance modification from a sparse set of inputs and novel view generation.

4 IMPLEMENTATION DETAILS

4.1 Optimizing TensoRF

The training/optimization of radiance fields requires information of the camera poses from which an image is captured. In the case of real scenes, we rely on COLMAP [32]to obtain this information and in the case of synthetic scenes, we use the data obtained from Blender. We optimize TensoRF (VM-Split-48) on the input images for $15k$ iterations. In each iteration, we shoot 4096 rays into the voxel grid. We obtain the radiance using volumetric rendering (Eq. 3) and optimize the grid iteratively.

$$C = \sum_{q=1}^{Q} \tau_q \left(1 - exp\left(-\sigma_q \Delta_q\right)\right) c_q,\ \tau_q = exp\left(-\sum_{p=1}^{q-1} \sigma_p \Delta_p\right) \quad (3)$$

We use Adam optimizer [22] which is initialized to a learning rate of 0.02 and is re-initialized to 0.02 after upsampling. The voxel grid is initialized with an effective resolution of 128^3 and iteratively upsampled every 1000 iterations, first upsampling starting 2000 until 5000 iterations are reached. We finally reach an effective voxel-grid resolution of 640^3 for real-world scenes and 300^3 for synthetic scenes. It is to be noted that the resolution mentioned here is *effective* but not exact, as the VM-decomposition provided by TensoRF presents a compact representation of voxel grid. Similar to TensoRF, we bi-linearly interpolate the matrix and linearly interpolate the vector in the VM decomposed representation during upsampling. This (bi-linear + linear) interpolation is similar to the tri-linear interpolation of the *full-voxel* grid. We perform such interpolations with neighboring voxels during the evaluation of a ray query. This enables us to obtain continuity in our rendered images. This pre-optimization phase requires 10-15 minutes. Once optimized it can be used to generate new views with various camera poses which post-stylization act as priors to our fine-tuning phase.

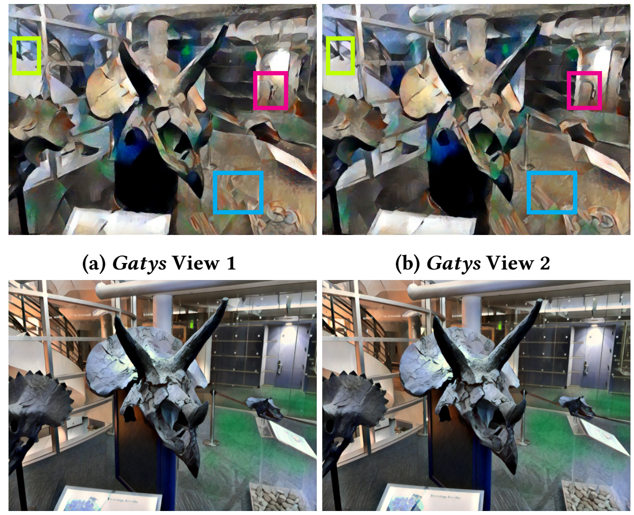
4.2 Stylization

Independently, we train the stylization module Johnson et al. [20] with the various reference style image on the COCO-14 Dataset [24] which takes 20 minutes. We train multiple such Johnson et al. [20] models for each desired style as the time required to train is quite less. We use this to create stylized priors from the views generated in the previous phase.

While optimizing for the style we freeze the density parameters in the TensoRF representation and optimize only the appearance parameters for a small number of iterations($1k$ iterations) with the stylized prior. This style adaption only takes a nominal time of 40-50 seconds. The style-adapted TensoRF representation obtained in the previous step can be thus used to generate novel stylized views using traditional voxel rendering techniques. The generation of each view takes around 4-5 seconds.

5 EXPERIMENTS

In this section, we present the various experiments with the proposed StyleTRF. We conduct all the experiments mentioned throughout the paper including the comparisons, on a workstation PC equipped with an AMD Ryzen-5800x and an NVidia RTX-3090 GPU. In Sec. 5.1 we discuss the choice of stylization module used to stylize the sparse prior used in our approach. We also experimentally compare and contrast between temporal-stylization of smooth trajectories of ground truth 3D content vs Actual 3D-Stylization in Sec. 5.2. Finally, in Sec. 5.3 we show the effect of different optimization strategies used to stylize the 3D content.



(a) *Gatys* View 1 **(b) *Gatys* View 2**

(c) *Johnson* View 1 **(d) *Johnson* View 2**

Figure 3: *Gatys vs Johnson Stylized Priors*: The figures shows the output of the stylization modules proposed by Gatys et al. [10] and Johnson et al. [20] on HORNS stylized based on *udnie.* Gatys et al. [10] produces inconsistencies in the style generated across near-by views which provide a poor prior to optimize appearance for our module. Johnson et al. [20] provides stable stylization across close-by views as seen in the figure.

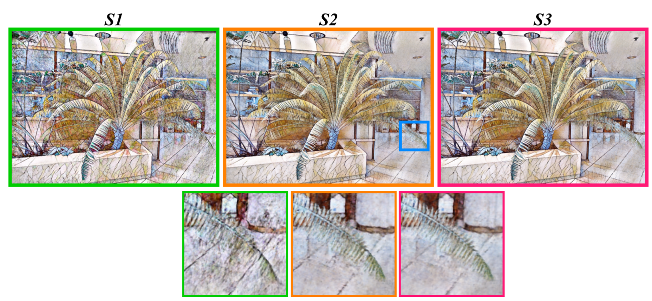


Figure 4: *Comparison of the different optimization strategies:* Strategy *S1* optimized directly on stylized images produces massive artifacts due to the loss in geometry. Unconstrained optimization *S2* on the other hand achieves results compared to *S1* while struggling to capture micro-details as seen in the insets. Our StyleTRF on the other hand freezes density while only optimizing for appearance, capturing the style while maintaining the geometric detail of the original scene.

5.1 Stylization Module

For stylizing the sparse prior required by our method we use Johnson et al. [20]. The concurrent work SNeRF[29] uses Gatys et al.

[10] method to iteratively stylize the radiance fields. We choose Johnson et al. [20] over Gatys et al. [10] because the latter depends on several factors such as the initialization, learning rate, and the optimization method which make it unreliable to obtain stable stylization across close-by views let alone temporal consistency. This can be observed in Fig. 3. This is one of the reasons, Nguyen-Phuoc et al. [29] requires multiple epochs to reflect the style in appearance.

On the other hand, Johnson et al. [20] use a fixed CNN-based architecture to infer the stylized image. Due to the spatial consistency of CNNs, two close-by views sharing significant spatial content lead to stable stylized close-by views.

5.2 Video Stylization v/s 3D Stylization

It can be argued that instead of stylizing 3D content, one can generate novel views on a camera trajectory and use temporally consistent stylization frameworks like ReReVST [38] to obtain stylized novel views.

However, we have observed that though temporal stylization is maintained, the work of ReReVST fails to fully capture the style information. The same can not be said for the stylization of 3D content. In Fig. 5 it can be seen that both StylizedNeRF and our method capture better style compared to ReReVST.

5.3 Optimization Strategies for Style Adaptation

For the stylization of Radiance fields we present three strategies:

(1) *S1:* Optimizing a TensoRF from scratch directly using the stylized priors.
(2) *S2:* Pre-optimizing a TensoRF on the original ground truth and adapting for style using sparse stylized priors *without freezing any parameters (both density and appearance).*
(3) *S3:* Pre-optimizing a TensoRF on the original ground truth data and adapting for style using sparse stylized priors while freezing the density.

When following the *S1* we observed geometric artifacts. This is because the stylized priors generated may not share the exact geometry with the ground truth. As stated above in Sec. 3.1, Johnson et al. [20] does not provide temporally consistent stylization which might also affect the geometry so-optimized in the stylized views in the case of *S1*.

Another strategy *S2* produces considerably better results compared to *S1* as seen in Fig. 4. This is because most of the geometric prior is learned from the pre-optimization phase which uses ground-truth images to obtain scene properties.

Though *S2* has produced good stylization at a micro-level, fuzzy geometry can be observed which reduces the appeal of the stylization. Specifically, thin geometric structures suffer from these undesirable fuzzy geometric changes. The limit-free optimization strategy of *S2* fiddles with the geometry components and leads to these artifacts. This can be observed in the insets provided in Fig. 4.

Our StyleTRF (*S3*) approach on the other hand alleviates these issues by freezing the geometric components of scene representation. Our approach generates crispier results compared to the rest of the aforementioned strategies *S1, S2*. This behavior is consistent across all the scenes.

We experiment with a different number of stylized priors to stylize the underlying scene. As shown in Fig. 7, we find that good stylization is obtained when all the priors cover the entirety of the scene. In most cases, 30-40 randomly sampled camera positions around the object suffice.

6 RESULTS

In this section, we compare our results against both 3D-stylization techniques and temporally-consistent stylization techniques both quantitatively and qualitatively.

6.1 Qualitative Results

For comparing qualitative results, we can use smooth trajectories which are similar to a video. This similarity enables us to compare with temporally-consistent stylization techniques alongside 3D-Stylization. For 3D stylized novel view synthesis, though there exists Huang et al. [17], they do not hold an accurate representation of geometry and rely on explicit geometry as input. Although other methods like Chiang et al. [7] do not rely on explicit geometry input, they suffer from patch border artifacts as discussed in [19] and Sec. 2. Hence, we compare our results with the latest 3D radiance field-based stylization method StylizedNeRF [19] and temporally-consistent stylization technique of ReReVST Wang et al. [38], on similar lines as [19].

Comparison amongst different stylization Techniques: We have observed that though the novel views generated using StylizedNeRF are reasonably consistent, the geometry after stylization in the case of StylizedNeRF has been greatly affected as seen in Fig. 5(c). It produces blurry geometry in the case of TRex stylized using *udnie* along with the missing greenish tints present in the *udnie*, and produces extremely noisy results for *mediterranean* applied on Fern . Contrary to this, our method distinctively transfers different colors to different parts of the scene as seen in TRex stylized using *udnie* and captures the complete color palette in *mediterranean* adapted onto Fern . Our method also consistently preserves geometry across all scenes and styles. The geometric noise in the StylizedNeRF can partly be attributed to the combined density and appearance encoded of a single MLP, which hinders the disentanglement of geometry and radiance as observed by [29].

In the case of the temporal-consistent style transfer technique ReReVST, we have observed that although the method is robust to adapt to new styles on the fly and produces results in real-time, it lacks proper capture of style information. This can be observed Fig. 5(b), ReReVST fails to capture the prevalent blue tint in *santa-maria* in the case of Flower , this could also be observed in the case of StylizedNeRF and vaguely captures the color palette of *mediterranean* in Fern . For *udnie* applied on TRex , it does not account for distinct colors present in the style and the resultant image contains an unappealing blend of colors throughout the image.

View Consistency: To qualitatively show view consistency, we render views across a smooth trajectory and show different views across it. In Fig. 6 we show our renders for *udnie* and *mosiac* styles adapted onto Stove and Horns respectively. It can be observed from the insets of Fig. 6 that stylized radiance across the frame is consistent, validating our claim that 3D-stylization-based novel view generation can produce multi-view consistent stylized content.

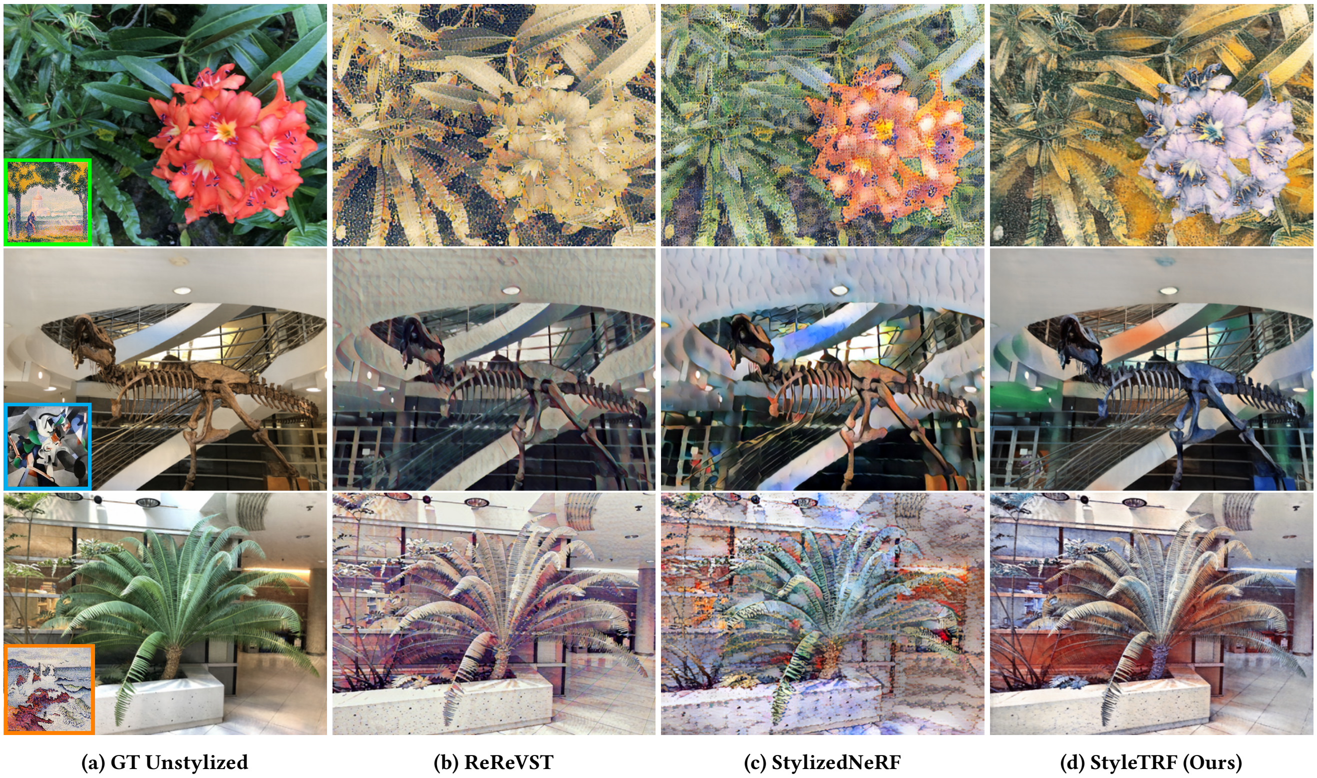


Figure 5: *Qualitative Comparison*: [Row 1: *santamaria*, Row 2: *udnie*, Row 3: *mediterranean*] Here we show the comparisons Unstylized frame in column-(a), temporally consistent stylization of ReReVST [38]in column-(b), StylizedNeRF [19] in column-(c) and stylization using our pipeline in column-(d). It can be observed that due to the usage of combined neural representation density and radiance, the style adaptation is affecting in the case of StylizedNeRF. Observe noisy geometric structures in *mediter* applied onto FERN

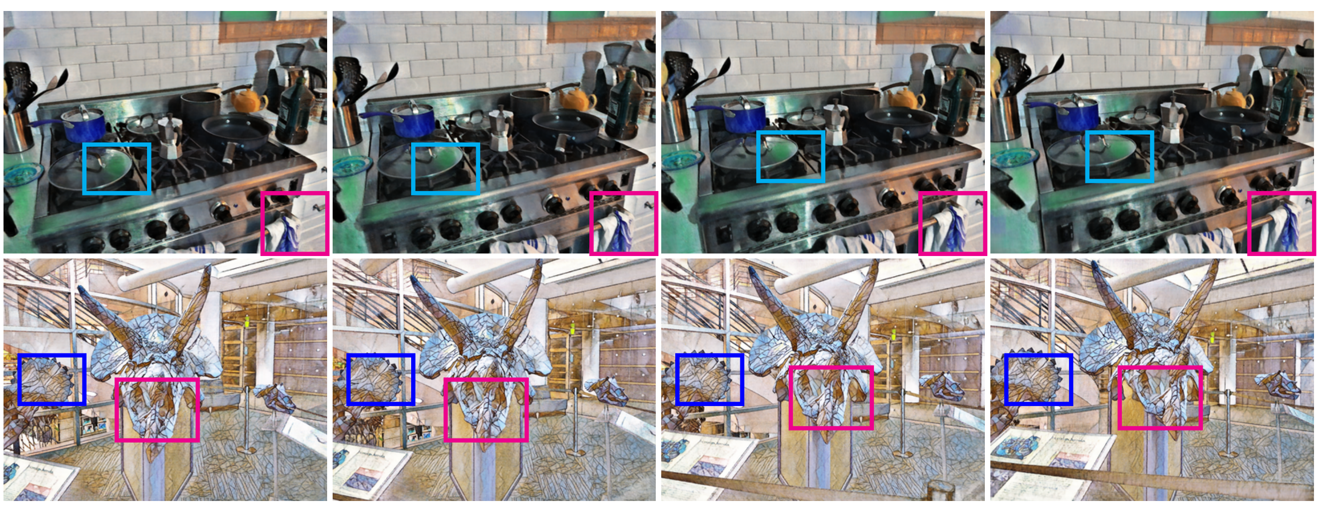


Figure 6: *View Consistency Across Frames*: The figure shows stylized novel views of a simple trajectory. To keep it simple we named the frames in t_i with increasing order of i from left to right. It can be observed clearly that our stylization is multi-view consistent both in the case of STOVE and HORNS .

Method / Scene	*ReReVST[38]*		*StylizedNeRF[19]*		*Ours*	
	short-term	*long-term*	*short-term*	*long-term*	*short-term*	*long-term*
Horns	0.0046	0.0137	0.0229	0.0239	**0.0040**	**0.0120**
Fern	0.0028	0.0080	0.0100	0.0168	**0.0020**	**0.0069**
Flower	0.0039	0.0106	**0.0020**	0.0277	0.0030	**0.0089**

Table 1: *Consistency Metrics:* We show *short-term* and *long-term* consistency metrics across a smooth trajectory generated using our appearance stylized scene representation. We have found that we obtain better *short & long-term* consistency compared to the SOTA 3D-Stylization technique StylizedNeRF[19], while maintaining better style transfer compared to temporal stylizing methods like ReReVST[38].

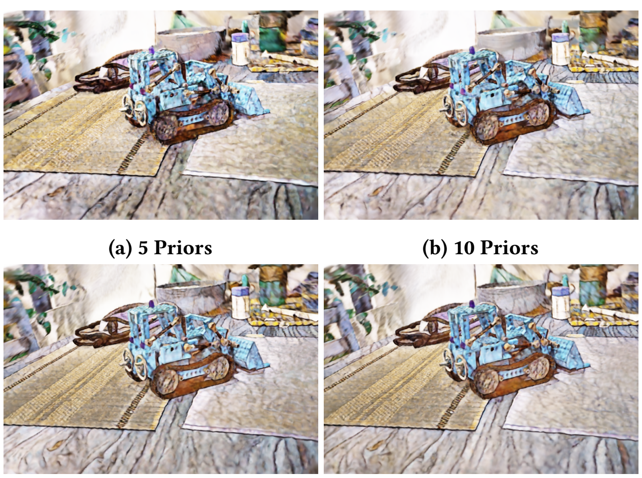
6.2 Quantitative Results

In order to check the consistency of our stylized content, we generate a smooth trajectory similar to SNeRF [29]. The rendered views along the trajectory are used to evaluate the consistency. Since the smooth trajectory replicates the behavior of a video we can comfortably compare our method with the temporal-consistent stylization techniques like ReReVST[38] alongside 3D-stylization methods like StylizedNeRF[19]. For this comparison, we chose Fern , Flower , and Horns scenes and report the respective metrics in Tbl. 1. We obtained better metrics compared to both temporal stylization [38] and implicit geometric stylization [19]. We estimated the consistency by calculating the optical flow O between the non-stylized frames $\mathcal{F}_i^{real}$ and $\mathcal{F}_{i+\delta}^{real}$ rendered using TensoRF.

$$O : opticalflow\left(\mathcal{F}_i^{real}, \mathcal{F}_{i+\delta}^{real}\right) \tag{4}$$

Using the so-obtained optical-flow O, we warp the stylized frame $\mathcal{F}_i^{style}$ to $\hat{\mathcal{F}}_{i+\delta}^{style}$.

$$\mathcal{W} : \hat{\mathcal{F}}_{i+\delta}^{style} \leftarrow Warp(\mathcal{F}_i^{style}, O) \tag{5}$$



(a) 5 Priors **(b) 10 Priors**

(c) 20 Priors **(d) 40 Priors**

Figure 7: *Number of Stylized Priors*: The images in this figure are obtained by fine-tuning StyleTRF using a different number of priors on the kitchen scene from the MipNeRF360 [2] dataset which is an unbounded 360 degree scene. Depending on priors quality improves and saturates at 35-40 priors.

For the calculation of optical flow, we use RAFT [36] similar to SNeRF.

We then calculate the pixel-averaged L_2 loss between the warped frame $\hat{\mathcal{F}}_{i+\delta}^{style}$ and the actual stylized frame $\mathcal{F}_{i+\delta}^{style}$. We aggregate this loss across the frame-combinations and report the metrics in Tbl. 1. For calculation of *short-term* consistency we chose $\delta = 1$ and for *long-term* $\delta = 5$. It is to be noted that δ here represents the change in the camera position along the trajectory. We have observed that though the *temporal-cosistent* stylization techniques like Wang et al. [38] come close to our results, they struggle to capture the reference-style (as seen in Fig. 5(b)).

	GT Training	Style Training	Style Adaptation
StyleImp	$\approx$ 12 hrs	NA	$\geq$ 5 hrs
SNeRF	$\approx$ 12 hrs	NA	3-4 days
StylizedNeRF	$\approx$ 12 hrs	NA	$\geq$ 3 hrs
Ours	20 mins	20 mins	45 secs

Table 2: *Training Times:* We compare our training times with StyleImp [7], SNeRF [29], StylizedNeRF [19]. Due to our compact representation, we get a training time of $\approx$ 20 mins compared to at least 12 hrs for others. The GT training and style training can be performed in parallel in our method. We disentangle the style training process from the style adaptation process which enables us to quickly adapt to any scene in under a minute.

7 CONCLUSIONS

In this paper, we presented StyleTRF, a compact and quick-to-optimize stylization technique which can generate stylized novel views of a scene. We have shown that our method can efficiently and faithfully incorporate style into a radiance field representation of a casually captured scene. We have qualitatively and quantitatively compared our method with the previous stylization methods. Our qualitative results and quantitative metrics demonstrate that StyleTRF is consistent across the views, having stylized the underlying 3D representation. We also reported the *short and long-term* consistency metrics which are better in most cases compared to the present 3D stylization methods. Concurrently, Zhang et al. [39] presented stylized novel view synthesis using voxel-based grids, partly similar to our method. But they chose PlenOxels [9] to represent the radiance fields. They focussed more on obtaining brush strokes while ours concentrates on geometric preserved fast-style adaptation.

Acknowledgements: Sirikonda Dhawal was partially supported by DST. We also acknowledge support from the TCS Foundation through the Kohli Center on Intelligent Systems (KCIS).

REFERENCES

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. 2020. Neural Point-Based Graphics. In *European Conference on Computer Vision (ECCV)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham.

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *Computer Vision and Pattern Recognition (CVPR)*. 5470–5479.

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch. 2020. NeRD: Neural Reflectance Decomposition from Image Collections. In *International Conference on Computer Vision (ICCV)*.

[4] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. 2021. Neural-PIL: Neural Pre-Integrated Lighting for Reflectance Decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[5] 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 (WACV)*.

[6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *European Conference on Computer Vision (ECCV)*.

[7] 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. In *Winter Conference on Applications of Computer Vision (WACV)*.

[8] Lieven De Lathauwer. 2008. Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness. *SIAM J. Matrix Anal. Appl.* 30, 3 (2008), 1033–1066. https://doi.org/10.1137/070690729 arXiv:https://doi.org/10.1137/070690729

[9] Fridovich-Keil and Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*.

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

[11] Pulkit Gera, Mohammad Reza Karimi Dastjerdi, Charles Renaud, P. J. Narayanan, and Jean-François Lalonde. 2022. Casual Indoor HDR Radiance Capture from Omnidirectional Images. In *British Machine Vision Conference (BMVC)*.

[12] Paris V. Giampouras, Athanasios A. Rontogiannis, and Eleftherios Kofidis. 2022. Block-Term Tensor Decomposition Model Selection and Computation: The Bayesian Way. *IEEE Transactions on Signal Processing* 70 (2022), 1704–1717. https://doi.org/10.1109/TSP.2022.3159029

[13] 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 (SIGGRAPH '96)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/237170.237200

[14] David Ha, Andrew Dai, and Quoc V. Le. 2017. HyperNetworks. In *International Conference on Learning Representations (ICLR)*.

[15] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. 2021. Baking Neural Radiance Fields for Real-Time View Synthesis. In *International Conference on Computer Vision (ICCV)*.

[16] 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 *Computer Vision and Pattern Recognition (CVPR)*.

[17] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. 2021. Learning to Stylize Novel Views. In *International Conference on Computer Vision (ICCV)*.

[18] Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *International Conference on Computer Vision (ICCV)*.

[19] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. 2022. StylizedNeRF: Consistent 3D Scene Stylization as Stylized NeRF via 2D-3D Mutual Learning. In *Computer Vision and Pattern Recognition (CVPR)*.

[20] 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 (ECCV)*.

[21] Petr Kellnhofer, Lars C. Jebe, Andrew Jones, Ryan Spicer, Kari Pulli, and Gordon Wetzstein. 2021. Neural Lumigraph Rendering. In *Computer Vision and Pattern Recognition (CVPR)*.

[22] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/ARXIV.1412.6980

[23] 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*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.).

[25] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. 2021. AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer. *International Conference on Computer Vision (ICCV)*.

[26] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. 2022. NeRF in the Dark: High Dynamic Range View Synthesis From Noisy Raw Images. In *Computer Vision and Pattern Recognition (CVPR)*.

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

[28] 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 *European Conference on Computer Vision (ECCV)*.

[29] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. 2022. SNeRF: Stylized Neural Implicit Representations for 3D Scenes. *ACM Trans. Graph.* 41, 4, Article 142 (jul 2022), 11 pages. https://doi.org/10.1145/3528223.3530107

[30] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *International Conference on Computer Vision (ICCV)*.

[31] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2016. Artistic Style Transfer for Videos. In *German Conference on Pattern Recognition (GCPR)*, Bodo Rosenhahn and Bjoern Andres (Eds.).

[32] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Computer Vision and Pattern Recognition (CVPR)*.

[33] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. 2018. Avatar-Net: Multi-scale Zero-Shot Style Transfer by Feature Decoration. In *Computer Vision and Pattern Recognition (CVPR)*.

[34] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. 2021. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *Computer Vision and Pattern Recognition (CVPR)*.

[35] Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. 2020. Two-Stage Peer-Regularized Feature Recombination for Arbitrary Image Style Transfer. In *Computer Vision and Pattern Recognition (CVPR)*.

[36] 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.

[37] Richard Tucker and Noah Snavely. 2020. Single-View View Synthesis With Multiplane Images. In *Computer Vision and Pattern Recognition (CVPR)*.

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

[39] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. 2022. ARF: Artistic Radiance Fields. In *European Conference on Computer Vision (ECCV)*.

[40] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. 2021. PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. In *Computer Vision and Pattern Recognition (CVPR)*.

[41] Kun Zhou, Yaohua Hu, Stephen Lin, Baining Guo, and Heung-Yeung Shum. 2005. Precomputed Shadow Fields for Dynamic Scenes. In *ACM SIGGRAPH 2005 Papers* (Los Angeles, California) *(SIGGRAPH '05)*. Association for Computing Machinery, New York, NY, USA, 1196–1201. https://doi.org/10.1145/1186822.1073332