ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446082

INSTANT PHOTOREALISTIC NEURAL RADIANCE FIELDS STYLIZATION

*Shaoxu Li, Ye Pan**

John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, China

ABSTRACT

We present Instant Photorealistic Neural Radiance Fields Stylization, a novel approach for multi-view image stylization for the 3D scene. Our approach models a neural radiance field based on neural graphics primitives, which use a hash table-based position encoder for position embedding. We split the position encoder into two parts, the content and style sub-branches, and train the network for normal novel view image synthesis with the content and style targets. In the inference stage, we execute AdaIN to the output features of the position encoder, with content and style voxel grid features as reference. The stylization of novel view images could be obtained with the adjusted features. Given a set of images of 3D scenes and a style target(a style image or another set of 3D scenes), our method can generate stylized novel views with a consistent appearance at various view angles in less than 10 minutes on modern GPU hardware.

Index Terms— Neural Radiance Fields, stylization

1. INTRODUCTION AND RELATED WORK

Image stylization has been widely researched since [1] presented a pioneering artistic style transfer algorithm. Fruitful follow-up works made it easier to adopt and higher quality for visualization[2, 3, 4]. Video stylization is a similar task to 3D scene stylization. Video stylization aims to style images and keep consistency between different frames[5, 6, 7].

For the task of 3D scene stylization, the methods vary with the 3D scene representation approach, such as mesh[8], point cloud[9], and NeRF[10]. In this paper, we focus on the NeRF stylization, whose inputs are image sets of scenes and outputs are novel view stylization images. NeRF[11] is a popular method for 3D scene representation. Some methods emerge with NeRF 3D scene stylization. Style3D[10] first attempted stylization for NeRF. Style3D[10] used an implicit representation of the 3D scene with the neural radiance fields model and a hypernetwork to transfer the style information into the scene representation. StylizedNeRF[12] proposed a mutual learning strategy for the stylized NeRF and 2D stylization method. StyleMesh[13] optimized an explicit texture for the reconstructed mesh of a scene and stylized it jointly from all available input images. ARF[14] propose a novel deferred back-propagation method to enable optimization of memory-intensive radiance fields using style losses defined on full-resolution rendered images. SNeRF[15] alternates the NeRF and stylization optimization steps. INS[16] decouples the ordinary implicit function into a style implicit module and a content implicit module to encode the representations from the style image and input scenes separately. UPST[17] proposed a hypernetwork to control the features of style images and use the 2D method to realize stylization. However, these methods made complex designs for composing 2D stylization methods, making them hard to extend. Besides, these methods suffer from a long training time.

To this end, we present Instant Photorealistic Neural Radiance Fields Stylization, a novel approach for multi-view image stylization for the 3D scene. Our method leverages the multi-resolution hash table architecture for NeRF proposed by Instant-NGP[18]. Our method uses the trainable feature vectors in the hash table to acquire the latent codes for stylization. First, we split the position encoder of instant neural graphics primitives into two parts: content and style. With the position encoder, our network can train for two scenes in less than 10 minutes. After training for normal scene synthesis, we execute AdaIN on position encoding features of novel view synthesis, with content and style voxel grid features as reference. Adjusted results are used for color predicting, and unadjusted results serve for density predicting. Unlike previous works whose style targets are only images, our method can accomplish stylization between two image sets for scenes. For stylization with images, we place the style images in the center of the 3D space and treat them as if they were scenes.

- We propose a novel Instant Photorealistic Neural Radiance Fields Stylization method. The training for novel view stylization image synthesis only costs 10 minutes.
- We propose, for the first time, style transfer from a 3D scene image set to a 3D scene image set. We extend the style target of NeRF stylization from images to 3D scene image sets.
- We propose to manipulate the position embedding features(with position encoder) of NeRF for stylization. Our method can be extended to the migration of more image stylization methods.

2. METHOD

The overview of our method has been shown in Fig 1. We train a model with two branches for content and style encoding in the training process. Then, we calculate the content and style position encoding features for voxel grid positions in the rendering process. With these features, AdaIN[19] is executed for stylization. The style target can be an image set for a scene or one style image.

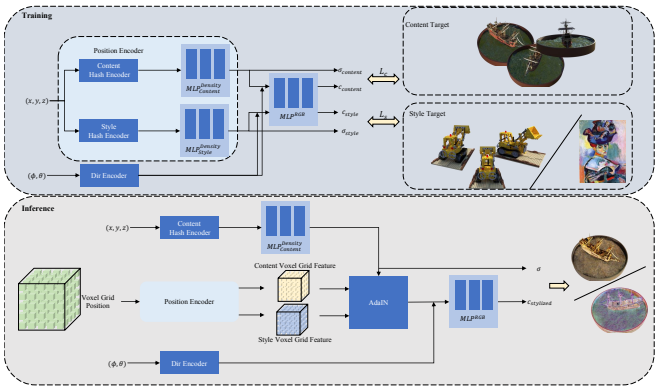

Fig. 1. The architecture of our method.

Hash encoder with multilevel grid feature vectors. For improving the training time of NeRF, many methods have been proposed. Instant-NGP[18] is one of state-of-the-art methods. Instant-NGP position encoder adopts a neural network with trainable weight parameters ϕ and trainable encoding parameters θ. Encoding parameters are arranged into L levels and conceptually stores feature vectors at the vertices of a grid. Parameters in each level contain up to T feature vectors with dimensionality. For a given input coordinate (x, y, z), the encoder searches the surrounding voxels at L resolution levels and the feature indices of these voxels by hashing their integer coordinates. For each level, the feature is the linear interpolation of the features according to the relative position of (x, y, z) and surrounding voxels. The encoded features are a concatenation of the result of each level. Compared with typical position embedding, the hash encoder works faster, which could speed up from hours to minutes, ensuring high-quality rendering.

2.1. Training with two parallel position encoders.

As an implicit neural representation, NeRF takes 3D coordinates x=(x,y,z) and views direction d as inputs, colors c=(r,g,b), and volume density as outputs. Usually, the implicit function is multilayer perception(MLP). Our method uses the hash encoder with multilevel grid feature vectors, an efficient architecture proposed by Instant-NGP. Unlike NeRF for a single scene, our network serves for two scenes. Our network divides the position encoder(hash encoder and $MLP^{Density}$) for a single scene into two branches and keeps other parts unchanged. The two branches accomplish the position embedding for the content and style scenes in the training process. Moreover, the direction embedding is accomplished by a shared direction encoder(spherical encoder). We train the network for novel view synthesis with content and style target simultaneously, as shown in Fig1.

When fitting scenes, we use Huber loss for both content and style targets:

$$L = \begin{cases} \frac{1}{2}rel^2, for\ rel <= \delta \\ \delta(rel - \frac{1}{2}\delta), otherwise. \end{cases} \qquad rel = |C_{predict} - C_{target}| \tag{1}$$

where $C_{predict}$, C_{target} denotes the rendering color value of images and the target value, δ is a hyperparameter. In the training process, The position encoders of content and style are optimized according to the content and style target, respectively. The MLP^{RGB} is optimized according to the content and style target.

For image set styles, we treat it as a regular 3D scene. The content and style scene are trained jointly. For a style image, we treat it as if it is placed in the center of the 3D space. We use 3D colored voxels to represent the 2D image pixels, as shown in Fig2. Rendered 3D colored voxel images with different views are used for training like a typical 3D scene. In the training process, random views are selected to optimize the radiance field.

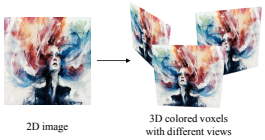

Fig. 2. 2D style image and corresponding 3D colored voxels with different views.

Compared with the network for one scene, our network has two position encoder sub-branches, which increases storage to a certain extent. The architecture of the hash encoder promises fast training for NeRF. Although we trained two scenes, our method's run time did not increase much. Our method could accomplish training in 10 minutes.

2.2. Stylization inference using AdaIN with voxel grid features.

Once trained, our network can render high-quality images for content and style scenes in novel views. We use AdaIN[19] for stylization inference in the inference stage. We first calculate the results of the content and style position encoder, with voxel grid positions as input. We use voxel grid $V^{N_V \times N_V \times N_V}$ to represent the 3D space, where N_V denotes voxel resolution. The results(Voxel Grid Feature) contain color and density features.

To maintain the geometry consistency of the scene, the density is reserved for the final rendering. For color stylization, an AdaIN is executed. Unlike the original AdaIN[19], our method executes AdaIN with reference parameters. The AdaIN module adjusts the feature $f(x, y, z)$ for stylization using Eq.2.

$$AdaIN(f(x,y,z)) = F^s_\sigma(\frac{f(x,y,z) - F^c_\mu}{F^c_\sigma}) + F^s_\mu \quad (2)$$

where F^c_μ, F^c_σ, F^s_μ and F^s_σ denote the mean and std of content and style voxel grid features. The AdaIN module adjusts the content position encoder features from the content distribution to the style target. According to the type of style target, our method can realize style transfer with artistic style images, photorealistic style images, and image set style images.

With AdaIN, the features are adjusted from content to style. In turn, the adjustment works from style to content features. Shared MLP^{RGB} promises the transfer is reasonable. Our method uses the feature outputs of the position encoder as latent codes for 3D scene stylization. Our method can be extended to more 2D stylization methods.

3. EXPERIMENTS RESULTS AND ANALYSIS

We conduct qualitative and quantitative experiments. Comparisons between our method and state-of-art methods illustrate the superiority of our method. In qualitative evaluation, we execute comparison on NeRF-Synthetic datasets[11] with artistic style images and Local Light Field Fusion(LLFF) datasets[20] with photorealistic style images. We demonstrate the results of style transfer between image sets of scenes. Moreover, we show the results of controlling the degree of stylization. In quantitative evaluation, we calculate the warped LPIPS metric[21] for short and long-term consistency. Furthermore, we conducted a user study on the LLFF datasets for stylization and consistency comparisons. The code is accomplished using Jittor[22] on a single Nvidia 3080 GPU.

3.1. Qualitative Results

Style transfer with artistic style images. In Fig.3, we demonstrate style transfer results with artistic style images. We compare our method with state of art methods UPST[17], INS[16], Style3D[10], Perceptual Loss[23], MCCNet[7] and ReReVST[24]. These comparison results are cited from [17]. Only INS, UPST and our method have good performances in geometry. Other results could not maintain high-quality results. For example, the whole space of the chair is cluttered in the result of Style3D. Artifacts exist with chair in the stylization results of Perceptual Loss ,MCCNet and ReReVST.

Style transfer with photorealistic style images. In Fig.4, we demonstrate style transfer results with photorealistic style images. We compare our method with state of art methods UPST[17], ARF[14], AdaIN[19], MCCNet[7], ReReVST[24]. These comparison results are cited from [17]. Our method, UPST and ReReVST preserve the scene's geometry. For color stylization, the details vary with methods.

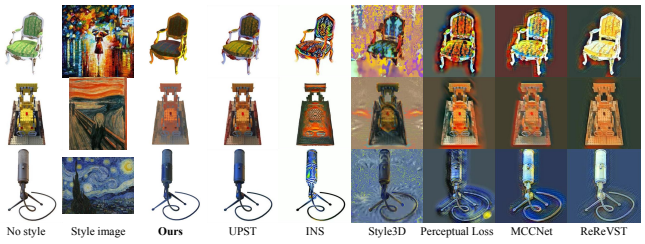

Fig. 3. Qualitative comparisons with artistic style images.

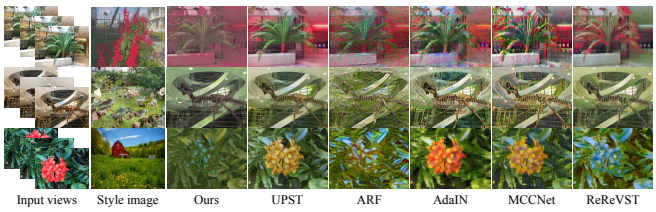

Fig. 4. Qualitative comparisons with photorealistic style images.

Style transfer between image set of scenes. Besides style images, our method is applicable to work with style transfer between image sets of two scenes. To the best of our knowledge, we are the first to propose style transfer between image sets of scenes. So, we only demonstrate some results without comparison in Fig. 5. Our method can accomplish stylization in the inference process from content to style and style to content.

Fig. 5. Stylization results of our method on NeRF-Synthetic datasets and LLFF datasets.

Content-style trade-off. The degree of stylization can be controlled in the inference by adjusting the style weight α in Eq.3. As shown in Fig.6, the style of images smoothly transfers between content and style targets by changing α from 0 to 1.

$$AdaIN_c(f(x,y,z)) = (1-\alpha)f(x,y,z)+\alpha AdaIN(f(x,y,z)) \quad (3)$$

Fig. 6. Content-style trade-off.

3.2. Quantitative Results

Consistency Measurement. We measure the short and long-term consistency using the warped LPIPS metric[21]. A view v is warped with the depth expectation estimated by NeRF. The score is formulated as:

$$E(O_i, O_j) = LPIPS(O_i, M_{i,j}, W_{i,j}(O_j)) \quad (4)$$

where W is the warping function, and M is the warping mask. Only pixels within the mask $M_{i,j}$ are taken for the calculation. Five scenes in the LLFF dataset are taken for comparison. We use 20 pairs of views for each scene and gap $5(O_i, (O_{i+5})$ and $15(O_i, (O_{i+15})$ for short and long-range consistency calculation. The short- and long-range consistency comparisons are shown in Tab. 1 and Tab. 2, respectively. Our method outperforms other methods by a significant margin.

Method	Fern	Flower	Horns	Orchids	Trex
AdaIN	0.0091	0.0077	0.0098	0.0099	0.0085
MCCNet	0.0075	0.0055	0.0088	0.0083	0.0070
ReReVST	0.0045	0.0030	0.0041	0.0060	0.0027
ARF	0.0055	0.0033	0.0061	0.0088	0.0064
UPST	0.0030	**0.0024**	0.0034	0.0041	0.0025
Ours	**0.0023**	0.0025	**0.0025**	**0.0037**	**0.0018**

Table 1. Short-range consistency.

Method	Fern	Flower	Horns	Orchids	Trex
AdaIN	0.0497	0.0458	0.0493	0.0353	0.0212
MCCNet	0.0407	0.0217	0.0270	0.0471	0.0168
ReReVST	0.0195	0.0110	0.0220	0.0394	0.0285
ARF	0.0355	0.0185	0.0247	0.0397	0.0218
UPST	0.0320	0.0100	0.0157	**0.0043**	0.0186
Ours	**0.0049**	**0.0099**	**0.0057**	0.0071	**0.0140**

Table 2. Long-range consistency.

User study. A user study compares the stylization and consistent quality of our method and others. We use the LLFF dataset for the study. Each scene in the LLFF dataset is stylized with our and other methods. We invited 50 participants and showed them the videos of stylization novel view synthesis results. For the same scene, we asked the participants to select the better results from our and another method, considering indicators stylization quality and geometry consistency. We collected 1000 votes for each indicator. Fig.7 shows the result of the study. Our method outperforms other methods in both stylization quality and geometry consistency.

Fig. 7. User study.

4. DISCUSSION

While our method shows competitive quality and better speed compared to state-of-the-art NeRF stylization methods, several challenges still need to be addressed in future work. The position and direction encoder output features are combined for color calculation. We only execute AdaIN on the position encoder features, leaving direction encoder features unchanged. So, our method does not handle stylization on the illumination and reflection.

Trainable encoding parameters are the core for fast scene training. For one scene, these parameters are random in twice training. Our method only promises the characteristics of stylization results from content to style. Nevertheless, the details of the results vary with the random encoding parameters. Finding a solution for fixed MLP^{RGB} for multi-scenes is necessary for stable results. If solved, the stylization could be accomplished with any two scenes using the fixed MLP^{RGB}.

5. CONCLUSION

In this paper, we present Instant Photorealistic Neural Radiance Fields Stylization, a novel approach that instantaneously styles 3D scenes. We split the position encoder of instant neural graphics primitives into two parts. This architecture lets our network train neural radiance fields for two scenes in less than 10 minutes. Due to the output features of the position encoder accounting for scene synthesis, it could be used for scene edit. We use AdaIN for scene stylization with voxel grid features, which could be extended to more image stylization methods. Our method extends the style target from style images to image sets of scenes. Our method can generate stylized novel views with a consistent appearance at various view angles, given a set of images of 3D scenes and a style target(a style image or another set of images of 3D scenes). Extensive experimental results demonstrate the validity and superiority of our method.

6. ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China(NSFC, NO. 62102255).

7. REFERENCES

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[2] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan, "Arbitrary style transfer with deep feature reshuffle," in *CVPR*, 2018.

[3] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," in *CVPR*, 2019.

[4] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang, "Attention-aware multi-stroke style transfer," in *CVPR*, 2019.

[5] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox, "Artistic style transfer for videos," in *German Conference on Pattern Recognition*, 2016, pp. 26–36.

[6] D. Chen, L. Jing, Y. Lu, N. Yu, and H. Gang, "Coherent online video style transfer," 2017.

[7] Yingying Deng, Fan Tang, Weiming Dong, haibin Huang, Ma chongyang, and Changsheng Xu, "Arbitrary video style transfer via multi-channel correlation," in *AAAI*, 2021.

[8] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[9] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang, "Learning to stylize novel views," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13849–13858.

[10] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu, "Stylizing 3d scene via implicit representation and hypernetwork," *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 215–224, 2021.

[11] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[12] Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao, "Stylizednerf: Consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning," in *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[13] Lukas Höllein, Justin Johnson, and Matthias Nießner, "Stylemesh: Style transfer for indoor 3d scene reconstructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 6198–6208.

[14] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely, "Arf: Artistic radiance fields," 2022.

[15] Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao, "Snerf: Stylized neural implicit representations for 3d scenes," *ArXiv*, vol. abs/2207.02363, 2022.

[16] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang, "Unified implicit neural stylization," *arXiv preprint arXiv:2204.01943*, 2022.

[17] Yaosen Chen, Qi Yuan, Zhiqiang Li, Yuegen Liu, Wei Wang, Chaoping Xie, Xuming Wen, and Qien Yu, "Upst-nerf: Universal photorealistic style transfer of neural radiance fields for 3d scene," in *arxiv*, 2022.

[18] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, July 2022.

[19] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *ICCV*, 2017.

[20] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, 2019.

[21] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[22] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou, "Jittor: a novel deep learning framework with meta-operators and unified graph execution," *Science China Information Sciences*, vol. 63, no. 222103, pp. 1–21, 2020.

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016, pp. 694–711.

[24] Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu, "Consistent video style transfer via relaxation and regularization," *IEEE Trans. Image Process.*, 2020.