StyleCity: Large-Scale 3D Urban Scenes Stylization

Yingshu Chen, Huajian Huang[†], Tuan-Anh Vu, Ka Chun Shum, and
Sai-Kit Yeung

The Hong Kong University of Science and Technology
`{ychengw,hhuangbg,tavu,kcshum}@connect.ust.hk, saikit@ust.hk`
[†]Corresponding author

Fig. 1: 3D city stylization in magic times of a day. Our proposed novel StyleCity framework can automatically stylize the textured meshes of large-scale urban scenes and generate harmonic omnidirectional sky backgrounds in a controllable manner with input style prompts, i.e., image and text references. Our effective solution has many potential applications, such as making city exploration in Google Earth a more personalized and visually captivating experience.

Abstract. Creating large-scale virtual urban scenes with variant styles is inherently challenging. To facilitate prototypes of virtual production and bypass the need for complex materials and lighting setups, we introduce the first vision-and-text-driven texture stylization system for large-scale urban scenes, StyleCity. Taking an image and text as references, StyleCity stylizes a 3D textured mesh of a large-scale urban scene in a semantics-aware fashion and generates a harmonic omnidirectional sky background. To achieve that, we propose to stylize a neural texture field by transferring 2D vision-and-text priors to 3D globally and locally. During 3D stylization, we progressively scale the planned training views of the input 3D scene at different levels in order to preserve high-quality

scene content. We then optimize the scene style globally by adapting the scale of the style image with the scale of the training views. Moreover, we enhance local semantics consistency by the semantics-aware style loss which is crucial for photo-realistic stylization. Besides texture stylization, we further adopt a generative diffusion model to synthesize a style-consistent omnidirectional sky image, which offers a more immersive atmosphere and assists the semantic stylization process. The stylized neural texture field can be baked into an arbitrary-resolution texture, enabling seamless integration into conventional rendering pipelines and significantly easing the virtual production prototyping process. Extensive experiments demonstrate our stylized scenes' superiority in qualitative and quantitative performance and user preferences. Project page: `https://chenyingshu.github.io/stylecity3d`.

Keywords: 3D Stylization · City Stylization · Neural Style Transfer

1 Introduction

Large-scale urban digital twins, reconstructed from satellite and UAV imagery, are widely utilized in various domains such as video games, 3D city visualization, and virtual reality applications. In virtual production, the demand for visually appealing and unique representations of reconstructed meshes is high. Yet, simulating the visual appearance of virtual scenes—like the time simulation in Google Earth—often fails to produce truly impressive visuals. This shortfall is typically due to the extensive need for laborious material design and meticulous illumination setups for each instance. To achieve automatically re-styling scene appearances, we seek to utilize 3D neural style transfer techniques.

Style transfer is a fundamental problem that seeks to automatically transfer the target style according to a given reference while maintaining the original structure. With the maturity of 2D neural stylization [14, 21, 29], 3D scene stylization has become more flexible and achievable by lifting content and style features priors from 2D to 3D. For example, image-guided mesh stylization [18] and text-guided object and room texturing [4, 23] realize 3D neural style transfer on meshes with 2D neural visual or textual priors. Due to the 2D-to-3D stylization fashion, we have to do view planning additionally to obtain comprehensive training views covering surfaces in balance [6, 37, 47]. However, these mesh stylization works primarily focus on object- and room-scale scenarios. Their view planning and style optimization methods are not suitable for large-scale scenes. Particularly, explicit texture or latent features optimization is resource-intensive. It results in affordable texture resolutions that are too limited to scale up to large-scale scenes. Moreover, these methods seldom address local semantic consistency in stylization. Given that large-scale scenes possess significant semantic complexity, we contend that maintaining semantic style feature consistency at a local level is crucial, particularly for achieving photo-realistic stylization. However, how to maintain global and local style harmony is challenging.

To narrow the gap, we propose to use a UV-based neural texture field to model scene appearances. The UV-based neural texture field is composed of a

multi-resolution feature grid for UV encoding and a multilayer perceptron (MLP) for feature decoding. As a compact hybrid representation, it reduces resource usage while maintaining global style consistency to some extent. After initializing the neural texture field with the original texture map, we perform optimization by minimizing global content and style losses among multiple planned views.

We employ a multi-scale progressive rendering strategy combined with scale-adaptive style reference enhancement during the texture field optimization to achieve high-fidelity texture renovation. The training views are progressively augmented into finer views at different levels with random translation and zoom-in. Simultaneously, we obtain multi-scale style reference features by randomly cropping the input reference image. Structural similarity scores guide us in adaptively selecting the scale of the style references to best match the scale of the current training views, ensuring the appropriate transfer of style patterns at different levels and providing precise and effective supervision. This approach allows us to progressively transfer style patterns to the 3D scene while maintaining fine-grained content structures.

In addition to global style optimization, our semantics-aware style loss preserves local style consistency, contributing to a holistic style enhancement. We also leverage a generative diffusion model to omnidirectionally denoise latent features and synthesize a visual and textual style-aligned omnidirectional sky image. Serving as the background of urban scenes, it provides a consistent and immersive atmospheric context for the stylized scene, despite having no illumination effect on the foreground during optimization and rendering. By converting the stylized neural texture field into a classical UV texture map, we efficiently render stylized novel views of the 3D scene in conventional pipelines for real-time applications. Fig. 3 illustrates the overview of our StyleCity framework.

To summarize, the contributions of this work include, 1) we propose a novel 3D textured mesh stylization framework using a neural texture field; 2) we introduce a multi-scale progressive rendering strategy and a scale-adaptive style optimization method to globally transfer 2D style to large-scale 3D scenes while preserving structural photorealism. 3) we propose a new local semantics-aware style loss to enhance local style correspondence; 4) We adopt the diffusion model to generate high-resolution 360-degree images as environment backgrounds enhancing the appeal of the stylized urban scene; 5) The comprehensive experiments prove that we realize high-quality 3D urban scene stylization without any manual rendering material and illumination setups. The automatic pipeline guided by vision and text references can be a quick substitute for virtual production and inspire scene design.

2   Related Work

In early works for 3D stylization, researchers applied interactive and iterative optimization on 3D data such as mesh texture and geometry to achieve non-photorealistic stylization [12,16,33,53]. Later, some methods [5,59] trained generative neural networks with large 3D data and facilitated 3D appearance and

geometry style transfer. For example, 3DStyleNet [59] transferred both shape and texture between 3D models with a textured mesh exemplar by a 3D generative model. 3D data collection is tedious and time-consuming. Fortunately, empowered by large-data pre-trained 2D neural models such as VGG, CLIP [45], and Stable Diffusion [48], there appeared a lot of automatic vision or text example-based 2D neural style transfer approaches [13,14,21,27,29,35,42,58]. Inspired by this, researchers tend to lift these 2D priors to 3D representations for automatic 2D example-based 3D stylization. Next, we review 3D neural style transfer methods on meshes and neural field representations, which transfer style with vision or text guidance and preserve to some degree original structure and geometry. We refer to [7] for more 3D stylization works on other 3D data.

Mesh-based Stylization. Kato *et al.* [25] proposed a neural renderer for mesh such that it enables artistic geometric and appearance stylization for the object mesh using image-example-based neural style transfer technique [14]. When it turns to image-guided 3D indoor scene artistic stylization, StyleMesh [18] introduced a texture optimization scheme specifically designed for 3D reconstructed indoor meshes, incorporating depth and angle information. Later, with the emergence of large-language model (LLM), text-guided stylization on mesh geometry and appearance [36,37], text-guided texture transfer or synthesis [6,24,31,47,57] became a popular topic in academia. [24] accomplished 3D semantic style transfer for indoor textures with text guidance. [36,37] achieved text-guided object geometric and texture stylization using a vertex-based neural style field network. They largely rely on fine-grained triangulation on the mesh and cannot synthesize detailed patterns or motifs. [31,57] took object materials into account, achieving text-guided texture stylization with material properties. With the advent of large-data pre-trained diffusion models [48], recent works [4,6,28,47,52,56] proposed to use the pre-trained diffusion models [3, 48, 61] to better supervise text-guided photorealistic texture synthesis for objects or rooms. [57] utilized 2D diffusion priors [44] for object texture stylization. However, globally optimizing explicit texture requires huge computational resources, i.e., GPU memory. These methods generally focus on small-scale scenes and lack scalability.

Neural Field Stylization. There appeared numerous neural-field-based stylization works for implicit reconstructed scenes [10,11,15,19,22,40,55,64]. Some data-driven generative training models [19,34] support zero-shot stylization for a scene, but require large training references and sophisticated model design. By contrast, feature optimized-based approaches usually have more impressive results without other style dilutions for one model [41,60,63]. For example, ARF [60] proposed a nearest-neighbor feature matching loss for local style matches in latent space transferring local artistic patterns. Empowered by 2D generative models, there are generative-model supervised optimization methods, such as InstructN2N [15], which utilized an alternative multi-view updating scheme to progressively optimize the 3D scene. However, multi-view supervision by independent one-shot generated images easily tends to collapse due to 3D inconsistency. These neural field stylization works utilized 3D neural field representations to maintain global style consistency and render novel views via
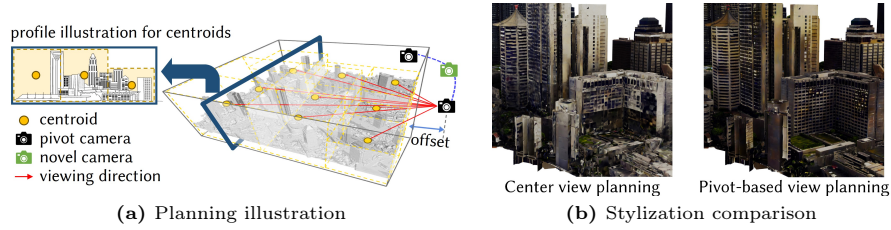
profile illustration for centroids

● centroid
📷 pivot camera
📷 novel camera
→ viewing direction

offset

(a) Planning illustration

Center view planning   Pivot-based view planning

(b) Stylization comparison

Fig. 2: Pivot-based view planning.

volume rendering. However, it is not wise to convert mesh-based rendering into volume rendering for mesh stylization which is prone to generating blurry views.

Compared to the aforementioned methods, our stylization system is the first work devised for mesh-based large complex outdoor urban scenes. Importantly, we explore the UV-based neural texture field to model mesh appearance and essentially take large-scale characteristics into account to devise style optimization strategies, achieving highly harmonious stylization in urban scenes.

3 Methodology

3.1 Data Pre-processing

To perform style transfer on a reconstructed mesh without known camera poses, it is indispensable to tackle view planning. In addition, we need to annotate 3D mesh with semantic labels such that we can efficiently obtain pixel-wise multi-view consistent semantic information during stylization.

Pivot-Based View Planning. Naive scene center view planning methods applied to stylize single objects [26, 37, 47] are insufficient for large-scale surface coverage. Direct application results in blurry texture with broken structure after stylization as shown in Fig. 2b. We developed a pivot-based view planning method that overcome their limitations. We uniformly sample P camera positions on the upper and side faces of the mesh bounding box with a proper offset. The mesh is then subdivided into r sub-regions, and the centroids of these regions are used as camera viewing points. Finally, we obtain $P \times r$ pivot views, locating at P pivot positions and looking at r centroids. Fig. 2a shows a pivot view planning example. Pivot views cover the majority of visible surfaces with associated semantics, serving as initialization for the novel views (Fig. 2a green camera) augmenting training, referred to Sec. 3.3 Multi-Scale Progressive Optimization.

Urban Scene Segmentation. For semantics-aware urban scenes stylization, we particularly consider classes of interest in architectural scenarios for segmentation, including "sky", "$building$", "$window$", "$road$", "$person$", "$plant$", "car", "$water$" and "$lights$". We adapted and fine-tuned Mask2Former [9], a pre-trained 2D semantic segmentation model, and implemented an automated 2D-to-3D segmentation tool via texture mapping for input 3D models. We also segment the input style reference image using the same segmentation model.

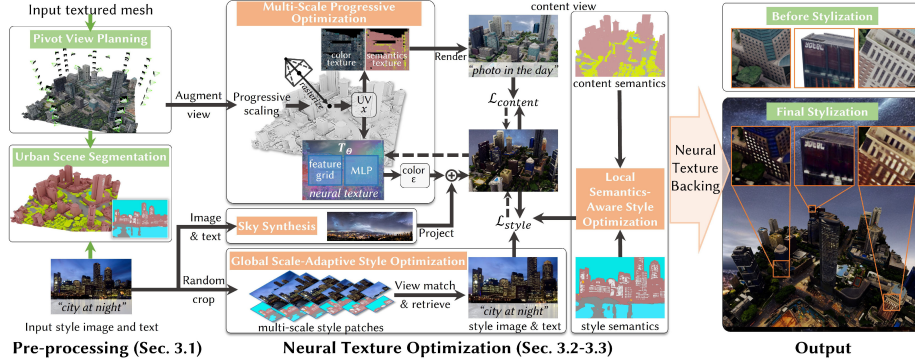Please refer to the supplementary for more data pre-processing details.

Fig. 3: Framework overview of StyleCity. Taking style image and text references as input, StyleCity optimizes the neural texture field in a semantics-aware fashion with progressively scaled training views. Synthesized omnidirectional sky enhances style atmosphere and assists semantic style supervision.

3.2   Neural Texture Field

Definition. Aiming at large urban scenes with megapixel or gigapixel texture images, we employ neural texture field representation and re-parameterize the huge texture map into a 2D continuous function. We define the neural texture field $T_\Theta(\cdot)$ (shown in *neural texture* in Fig. 3) by an MLP that maps a normalized UV texture coordinate ($x \in \mathbb{R}^2$) with hash grid-based feature encoding [39] to a color RGB value ($\varepsilon \in \mathbb{R}^3$): $T_\Theta(x) = \varepsilon$. The neural texture field theoretically supports arbitrary texture resolution with a condensed amount of parameters. It is compact but sufficient to represent scene appearance with on average 90% texture size compression from experiments. Moreover, neural continuous representation learns smooth interpolation among UVs, tolerating inevitably incomplete training views and maintaining global style consistency. By contrast, explicit texture optimization may lead to an out-of-memory issue with high resolution or a degraded discrete stylized texture with insufficient resolution.

Neural Rendering. Given a camera pose, we rasterize mesh and retrieve UVs to query texture model $T_\Theta(\cdot)$. Accordingly, we obtain corresponding texture RGB values, and then resemble values into a rendered image.

Conventional Rendering. To get stylized texture with width W_{tex} and height H_{tex}, we query $T_\Theta(UV)$ with $U \in [0, 1]$ with W_{tex} uniformly sampled values and $V \in [0, 1]$ with H_{tex} uniformly sampled values. The output texture with mesh can be rendered via the conventional rendering pipeline or applied to VR.

Initialization. Before stylization, we distill original full-resolution texture content to neural texture by a simple MSE photometric loss $\mathcal{L}_{tex} = \sum_{x \in UV} \| T_c(x) - T_\Theta(x) \|_2^2$, which T_c is original texture image, x is a random sampled batch of UVs. After distillation, the neural texture is initialized with original texture style.

3.3   Content and Style Joint Progressive Optimization

We jointly optimize the neural texture field through multiple views with both source content and target style constraints in each iteration. As illustrated in Fig. 3, we sample a viewpoint, render the content view and its segmentation, and obtain the stylized view from neural texture. To achieve high-quality stylization, we augment training views into multi-scales for progressive optimization.

Multi-Scale Progressive Optimization. During optimization, we randomly sample novel views along Bezier curves with nearby planned pivot cameras as control points in order to enlarge covered angles, as shown in Fig. 2 green camera. Furthermore, we randomly translate novel view cameras horizontally and vertically in an offset of $[-\alpha, \alpha] \times dist_{cam}$, where $dist_{cam}$ is the distance from camera to model center, α is hyperparameter (default 0.25). However, although the dynamic view augmentation ensures each surface is comprehensively stylized and globally harmonic, we found that the stylized mesh still lacks high-frequency details. To address this issue, we progressively scale sampled views at a preset field of view (FoV) with a "zoom-in" effect during training, as demonstrated in Fig. 4.

Specifically, FoV will decrease uniformly (e.g. from 90° to 20°) while the number of novel sampling views increases at each progressive optimization level. We set all rendered images in a fixed resolution and the default number of levels N=5. The experiment validates that dynamic multi-scale progressive optimization is effective for high-fidelity large-scale scene stylization. For simplicity, we only demonstrate the optimization process

Fig. 4: Multi-scale training views augmented based on same pivot cameras.

and loss equations for a single view at any optimization level in the subsequent description, unless explicitly stated otherwise.

Content and Photorealism Preservation Optimization. During stylization, we utilize not only content features but also Laplacian values of rendered views for supervision, preserving texture content and scene identity. During training, we mask out sky background areas.

Latent content feature loss can help preserve the basic appearance of the original texture, such as original colors and abstract structures, while remaining possible for style variations: $\mathcal{L}_c = \sum_{l \in l_c} \|F^l(c) - F^l(z)\|_2^2$. $F^l(\cdot)$ denotes embedded features of lth layer in VGG-19 [50] taking an image as input. c and z are rendered content image and stylized image on the current viewpoint respectively, the feature layer $l_c = \{4\}$ is used.

Beyond the content feature loss, we incorporated a photorealism regularizer, akin to the approach in [35], to maintain high content fidelity and preserve

identity, such as by maintaining texture edges irrespective of chrominance. The regularization term is formulated as $\mathcal{L}_{pht} = \sum_{h=1}^{3}(\mathcal{V}_h^z)^T\mathcal{M}^c\mathcal{V}_h^z$, where h denotes a RGB channel, and $\mathcal{V}^z$ is the flattened version of the stylized view. The kernel $\mathcal{M}^c$ is the matting Laplacian matrix [32] of the content view.

Global Scale-Adaptive Style Optimization. Global style optimization is responsible for globally transferring style features to the neural texture field for overall atmosphere alignment.

As described before, our multi-scale progressive optimization strategy renders different-scale images at different training levels. To keep the style consistent, the style reference image should adapt to the scale of the training views. It means that style reference features should be scale-adaptive. To this end, we exploit patch-based style references at different resolutions for optimization. During optimization, we randomly crop the input style image s into multi-scale style patches S with different dimensions. The long side of each style patch is no shorter than 256 pixels. Then we select a structure-matched style patch $\hat{s}$ for each training view c, as illustrated in Fig. 5. Inspired by [49, 54], we apply a self-similarity
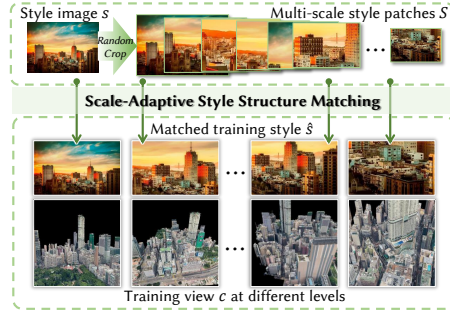

Fig. 5: Scale-adaptive style structure matching. We match multi-scale styles based on structure similarity between training views and style patches.

descriptor derived from latent features to represent spatial structure. The training style patch $\hat{s}$ is selected by minimizing the structure disparity between the training view c and the style patch:

$$\arg\min_{\hat{s}\in S} \sum \|\mathcal{D}(c) - \mathcal{D}(\hat{s})\|_2^2, \tag{1}$$

where $\mathcal{D}(\cdot) = sim(F^{l_d}(\cdot), F^{l_d}(\cdot)^T)$ is the self-similarity structure descriptor derived from channel-wise flattened VGG features, $l_d = \{3\}$ for simplicity, and $sim(\cdot, \cdot)$ is the cosine similarity. With scale-adaptive style matching for optimization, we can better guarantee sufficient and appropriate style feature transfer.

Global Vision-and-Text Driven Style Loss. To quickly grab and generate meaningful new stylized texture, we deliver global multi-level visual style features to texture, by penalizing the discrepancy of global style feature distributions: $\mathcal{L}_{gs} = \sum_{l\in l_s}(\|\mu(F^l(\hat{s})) - \mu(F^l(z))\|_2^2 + \|\sigma(F^l(\hat{s})) - \sigma(F^l(z))\|_2^2)$, where $\hat{s}, z$ respectively denote the matched style image and output stylized image; feature layers $l_s = \{2, 3, 4\}$; μ and σ indicate mean and standard deviations.

To transfer textual style features for holistic style semantics harmony, we apply global and directional CLIP losses [13, 45]:
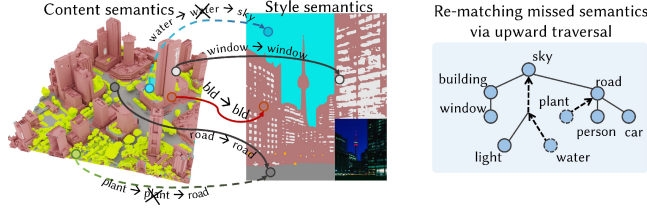
Fig. 6: Semantics matching (left) and hierarchy graph for semantics re-matching (right graph). Style reference lacks semantics "*water*" and "*plant*" contained in the content scene. During semantic stylization, these semantics are re-matched to upper-level existing semantics in style reference.

$$\mathcal{L}_{clip} = \mathcal{L}_{glb} + \mathcal{L}_{dir},$$
$$\mathcal{L}_{glb}(s_t, z) = 1 - sim(E_T(s_t), E_I(z)), \mathcal{L}_{dir}(c, z, c_t, s_t) = 1 - sim(\Delta I, \Delta T), \quad (2)$$

where $sim(\cdot, \cdot)$ is cosine similarity, c_t and s_t are source and style text prompts. E_I and E_T are image and text encoders, $\Delta I = E_I(z) - E_I(c), \Delta T = E_T(s_t) - E_T(c_t)$.

Local Semantics-Aware Style Optimization. For urban scenes with complex contexts, global style transfer easily leads to mismatches of style semantics. Thus, we introduce a local style optimization strategy for class-wise feature regularization to achieve more realistic stylization.

Local Semantics-Aware Style Loss. The local semantics-aware optimization is performed by minimizing the mean and standard deviation of class-wise style features between the rendered stylized view z and style reference s. We render a corresponding semantic segmentation map for the current training view using the semantics texture of the mesh while retrieving a semantic segmentation map for the matched style reference image. Based on semantic segmentation maps, we sample region of interest (ROI) features of each class for the training view and style reference image respectively:

$$f_i^l(z) = ROI_i(F^l(z)), \; f_i^l(s) = ROI_i(F^l(\hat{s})), \quad (3)$$

where $ROI_i(\cdot) \in \mathbb{R}^{1 \times M_i}$ indicates flattened vector of regions of ith class, M_i is the number of non-zero value in the binary mask of ith class; $F_i^l(\cdot)$ is the lth layer VGG features in i^{th} class regions, $\hat{s}$ is structure-matched style image. Different from [35] which introduced style augmentation using masked Gram matrix taking non-ROI area into account, our semantic style loss only calculates the style feature of the target area representing a more accurate feature distribution. Moreover, the ROI features have different dimensions among different views. With N_{cls} classes of interest, we formulate local semantics-aware style as:

$$\mathcal{L}_{ls} = \sum_{l \in l_s} \sum_{i=1}^{N_{cls}} (\|\mu(f_i^l(s)) - \mu(f_i^l(z))\|_2^2 + \|\sigma(f_i^l(s)) - \sigma(f_i^l(z))\|_2^2). \quad (4)$$

Semantic Hierarchical Re-match. During style transfer, it may occur mismatched and missed semantics between content view and style image as shown in Fig. 6. Content scene contains semantics "*building*", "*window*", "*plant*", "*water*" and "*road*", while style image lacks corresponding semantics such as "*plant*" and "*water*". To obtain references for missed semantics, we designed a hierarchy graph for semantics re-matching based on illumination effects in the real world. For example, cars, trees, and people on the road should share similar illumination of the road; all elements have the same ambient illumination the sky, etc. In Fig. 6, "*water*" content can match "*sky*" style luminance, "*plant*" content can match "road" reference luminance. If the matched style patch fails to provide style of some classes, we turn back to full-scale style reference for help.

Total Objective. We now have the texture stylization model with all losses:

$$\mathcal{L}_{total} = \lambda_c\mathcal{L}_c + \lambda_{pht}\mathcal{L}_{pht} + \lambda_{gs}\mathcal{L}_{gs} + \lambda_{ts}\mathcal{L}_{clip} + \lambda_{ls}\mathcal{L}_{ls}. \tag{5}$$

By default, we set $\lambda_c = 10, \lambda_{pht} = 1e-3, \lambda_{ls} = 1e-1, \lambda_{gs} = 1, \lambda_{ts} = 5$, and optimize each model for 20 epochs in 10 hours. Particularly, we set $\lambda_{pht} = 0$ in the first 2 epochs to get basic style patterns in case of style suppression from $\mathcal{L}_{pht}$ in the beginning.

3.4 Style-Aligned Omnidirectional Sky Synthesis

Given the same style image and text references, we adopt the pre-trained Stable Diffusion model [48] and conduct vision-and-text aligned sky panorama synthesis. A style-aligned sky, serving as the unique background of urban scenes, is essential to establish a global atmosphere for the final rendering. Moreover, the sky background supports comprehensive visual semantics, enhancing the effectiveness of textual style loss (Eq. 2). The pre-trained diffusion model excels in generating small image patches (e.g., 512×512), while diffusion panorama synthesis methods [3,30] utilize multi-window joint diffusion technique to generate high-resolution panorama. However, to support rendering with an environment texture, we require a high-resolution sky image covering 360-degree FoV. Due to uniform diffusion among sliding windows, the panorama is generally flattened and suffers layered content when diffusing latent features vertically, such as the SyncDiffusion [30] results demonstrated in Fig. 7b. To achieve high-resolution panorama synthesis as a whole entity, we extend the multi-window joint diffusion technique and propose omnidirectional latent feature sampling for view supervision. At each denoising step, we leverage a 90-degree bounding field of view (BFoV) [20] and omnidirectionally sample the latent features to extract noisy latent feature sub-regions. Each sub-region latent feature is then denoised by the denoising diffusion implicit model (DDIM) [51] and decoded by the variational autoencoder (VAE) decoder to generate a color image. Meanwhile, we apply the perceptual similarity constraint $\mathcal{L}_{LPIPS}$ [62] between the input style reference and generated images to supervise similar visual style. Fig. 7a illustrates the sky synthesis process; Fig. 7b and 10 display comparison and synthesis examples. Please refer to the supplementary for more details and comparisons.
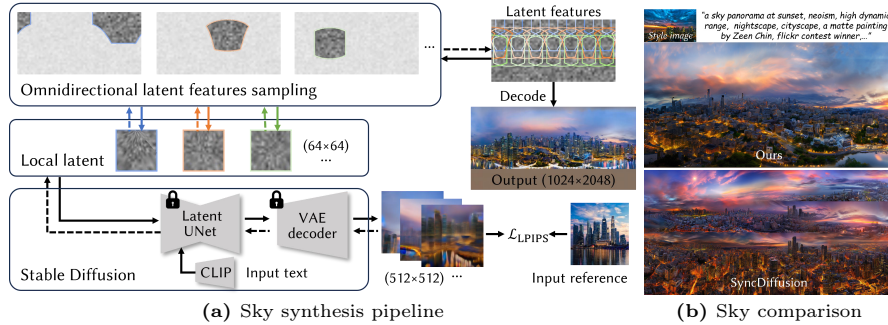
(a) Sky synthesis pipeline   (b) Sky comparison

Fig. 7: (a) High-resolution sky synthesis with omnidirectional sampling. (b) Panoramic sky synthesis comparisons with SyncDiffusion [30] using same seeds and text prompts.

4   Experiments

We conducted experiments on UV textured models in different cities collected by Google 3D Tiles API [2]. For evaluation, we rendered a sequence of testing views around each model. For 2D style images, we referred to city-view evaluation images in [8, 35], and applied the text-guided synthesis model [61] to generate some style images given customized style text prompts. Source texts follow templates such as "a photo in the day", and non-customized target texts are generated by [43] given style images. All experiments were conducted on RTX 3090 GPU. We implemented our method based on Pytorch and Pytorch3D [46], and used tiny-cuda-nn [38] for multi-resolution grid-based hash encoding and network training acceleration. Resolution 512×512 was used for optimization.

Result Comparisons. Given the downgraded performance of image or video stylization in 3D scenarios regarding 3D geometric and appearance consistency [10, 19, 22], our focus lies on 3D stylization baselines. We compared our method with the image-guided explicit texture stylization StyleMesh [18], image- and text-guided NeRF stylization methods ARF [60] and InstructN2N [15]. We ran baselines using official source codes and increased the content loss hyperparameter used in StyleMesh and ARF by 10 times for photorealism. We took the same pivot views (Sec. 3.1) as their training input. We evaluated the performance of foreground content similarity by edge-SSIM (eSSIM) and masked LPIPS [62] between source and stylized testing views, and semantic style similarity by image-text CLIP score [17] between the text reference and stylized testing views with the same sky background. Moreover, we conducted a user study on content preservation and style matching of stylized models. Table 1 and Fig. 8 demonstrate quantitative and qualitative comparisons.

By comparison, our results achieved higher fidelity of motifs and style details such as logos, and semantically matched styles like lit windows, while other baselines lead to blurriness or non-photorealism. The 2D generative diffusion-based 3D lifting method InstructN2N tends to have minor global color changes although preserves more original content features. We also found it difficult

Table 1: Quantitative comparisons. We evaluated 5 city models and 7 styles, resulting in 35 stylized outputs for each method. These city models, spanning approximately 0.5km × 0.5km in physical area, represent diverse locations worldwide, including London, Sydney, New York, Hong Kong, and Singapore.

Method	3D Representation	Condition	Scalability	eSSIM↑	LPIPS↓	CLIP↑	User Study (score in 0-5)	
							Content Preserve↑	Style Match↑
ARF [60]	NeRF	image	object-centric	0.394	0.812	0.188	1.576	1.430
StyleMesh [18]	explicit texture+mesh	image	room-level	0.400	0.576	0.188	2.074	2.402
InstructN2N [15]	NeRF	text	object-centric	0.503	0.370	0.186	3.156	2.024
Ours	neural texture+mesh	image+text	urban-level	**0.623**	**0.342**	**0.244**	**4.586**	**4.572**

Fig. 8: Visual comparisons with baselines. Zoom in for details.

to learn abstract-style prompts. For example, it always coats the scene with blue for "blue hour" prompts even along with other style-related descriptions. Optimization-based artistic style transfer methods (StyleMesh and ARF) lead to dramatic content distortion without photorealism concerns. From visual results, hazy artifacts appear in all scenes in volume rendering based neural field stylization (ARF and InstructN2N). From the user study's feedback, we found ARF got the lowest "style match" score because of its blurry appearance. Besides, in the absence of semantic matching, image-guided methods tend to transfer mismatched styles, such as ARF stylizing buildings with the purple sky and background with building colors in Fig. 8.

Ablation Study. Fig. 9 shows comparisons for ablation studies. We randomly pick a scene with 4 different styles for quantitative evaluation.

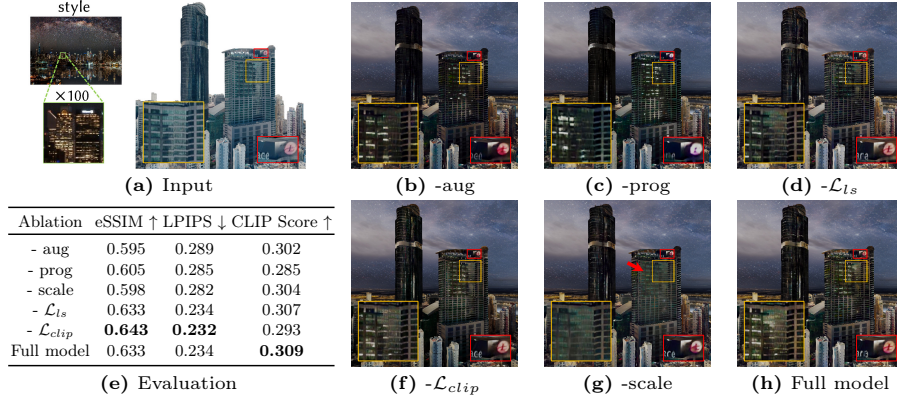| Ablation	eSSIM ↑	LPIPS ↓	CLIP Score ↑
- aug	0.595	0.289	0.302
- prog	0.605	0.285	0.285
- scale	0.598	0.282	0.304
- $\mathcal{L}_{ls}$	0.633	0.234	0.307
- $\mathcal{L}_{clip}$	**0.643**	**0.232**	0.293
Full model	0.633	0.234	**0.309**

(e) Evaluation

Fig. 9: Ablation studies. Without (b) view augmentation, (c) progressive optimization or (g) scale-adaptive styles, it may perform worse on sharp content preservation. Without (d) local style loss, (f) CLIP loss or (g) scale-adaptive styles, it failed to perform some detailed effects, such as lit windows.

Effect on augmented view planning and progressive optimization. Augmentation with novel view sampling and view translation (*aug*), and multi-scale progressive optimization (*prog*) assist in high-fidelity content structure preservation. As shown in Fig. 9b and 9c, without either of them the system only restores rough textures of blurry logos and windows with worse eSSIM and LPIPS performance. *Effect on semantics style loss.* Without local semantics-aware style loss ($\mathcal{L}_{ls}$) or textual semantic CLIP loss ($\mathcal{L}_{clip}$), the system generates fewer detailed novel patterns such as lit windows (Fig. 9d and 9f). It leads to fewer style changes but may improve original content similarity, e.g., slightly better eSSIM and LPIPS. *Effect on scale-adaptive style optimization.* Scale-adaptive style references help with robust stylization. In some challenging scenarios, unmatched-scale style can dull colors and even destroy geometry structure. For example, the subtle window patterns in the style image compromised geometric definition and the synthesis of lit windows, as shown in the area highlighted by the arrow in Fig. 9g.

Applications. Our proposed system enables users, especially non-experts, to have the option to personalize their 3D mesh models by selecting different reference images or text prompts. For example, the system can be used to "imitate" *(1) Time-of-Day Effect* shown in Fig. 1. We can tailor the visual appearance of the virtual environments to our preferred time, creating a more personalized and engaging experience. Our system also supports *(2) Non-photorealistic Stylization.* For artistic reference, we optimize texture without initialization and set $\lambda_{pht} = 0, \lambda_{gs} = 10, \lambda_{ls} = 1$. In addition, StyleCity supports *(3) 2D-to-3D Style Editing and Propagation.* This feature allows users to apply any 2D stylization method, such as image- or text-guided style editing [61] and manual editing, to edit a 2D view s_{edited} at a specific viewpoint of the scene. To ensure consistency, during optimization we additionally employ the penalty term $\mathcal{L}_{edited} = \|s_{edited} - z_{edited}\|_2^2$ to the stylized image z_{edited} at this viewpoint, and

(a) Input   (b) Stylized results

Fig. 10: More stylization results of our proposed StyleCity. In the second-to-last row, a style reference from the "Cyberpunk 2077" video game [1] demonstrates a result for futuristic style transfer. The last row displays the artistic 2D-to-3D editing result, and it tends to generate novel textual semantic patterns unseen from image reference.

use the edited image s_{edited} as the style reference. The last two rows in Fig. 10 display examples of the futuristic style of "Night City" in Cyberpunk 2077 [1] and 2D-to-3D editing in the artistic "Neo-Impressionism" style.

5 Conclusion

We present StyleCity, a vision-and-text driven stylization pipeline for large-scale city-level textured mesh stylization. We utilize a neural texture field to model the scene appearance and propose a novel multi-scale progressive optimization method achieving high-fidelity stylization. For harmonic stylization, we introduce scale-adaptive style optimization and new loss functions to regularize style features globally and locally. Besides, we improve the diffusion panorama synthesis method to support style-aligned high-resolution omnidirectional sky synthesis, which serves as background for an immersive atmosphere and better semantics supervision. Finally, the output-baked texture supports conventional rendering in real time. Our system originally solves challenges in urban scene stylization in high fidelity and a controllable manner, potentially facilitating various applications including virtual production.

Acknowledgement

References

1. Cyberpunk 2077 (2023), `https://www.cyberpunk.net/`
2. Google maps tile api (2023), `https://developers.google.com/maps/documentation/tile/`
3. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: Fusing diffusion paths for controlled image generation. In: Proceedings of the 40th International Conference on Machine Learning (2023)
4. Cao, T., Kreis, K., Fidler, S., Sharp, N., Yin, K.: Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4169–4181 (2023)
5. Cao, X., Wang, W., Nagao, K., Nakamura, R.: Psnet: A style transfer network for point cloud stylization on geometry and color. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3337–3345 (2020)
6. Chen, D.Z., Siddiqui, Y., Lee, H.Y., Tulyakov, S., Nießner, M.: Text2tex: Text-driven texture synthesis via diffusion models. In: roceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
7. Chen, Y., Shao, G., Shum, K.C., Hua, B.S., Yeung, S.K.: Advances in 3d neural stylization: A survey. arXiv preprint arXiv:2311.18328 (2023)
8. Chen, Y., Vu, T.A., Shum, K.C., Hua, B.S., Yeung, S.K.: Time-of-day neural style transfer for architectural photographs. In: 2022 IEEE International Conference on Computational Photography (ICCP). IEEE (2022)
9. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1290–1299 (2022)
10. Chiang, P.Z., Tsai, M.S., Tseng, H.Y., Lai, W.S., Chiu, W.C.: Stylizing 3d scene via implicit representation and hypernetwork. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1475–1484 (2022)
11. Fan, Z., Jiang, Y., Wang, P., Gong, X., Xu, D., Wang, Z.: Unified implicit neural stylization. In: European Conference on Computer Vision. pp. 636–654. Springer (2022)
12. Fišer, J., Jamriška, O., Lukáč, M., Shechtman, E., Asente, P., Lu, J., Sỳkora, D.: Stylit: illumination-guided example-based stylization of 3d renderings. ACM Transactions on Graphics (TOG) **35**(4), 1–11 (2016)
13. Gal, R., Patashnik, O., Maron, H., Bermano, A.H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. ACM Transactions on Graphics (TOG) **41**(4), 1–13 (2022)
14. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)

15. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
16. Hauptfleisch, F., Texler, O., Texler, A., Krivánek, J., Sỳkora, D.: Styleprop: Real-time example-based stylization of 3d models. In: Computer Graphics Forum. vol. 39, pp. 575–586. Wiley Online Library (2020)
17. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: CLIPScore: a reference-free evaluation metric for image captioning. In: EMNLP (2021)
18. Höllein, L., Johnson, J., Nießner, M.: Stylemesh: Style transfer for indoor 3d scene reconstructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6198–6208 (2022)
19. Huang, H.P., Tseng, H.Y., Saini, S., Singh, M., Yang, M.H.: Learning to stylize novel views. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13869–13878 (2021)
20. Huang, H., Xu, Y., Chen, Y., Yeung, S.K.: 360vot: A new benchmark dataset for omnidirectional visual object tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20566–20576 (2023)
21. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017)
22. Huang, Y.H., He, Y., Yuan, Y.J., Lai, Y.K., Gao, L.: Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18342–18352 (2022)
23. Hwang, I., Kim, H., Kim, Y.M.: Text2scene: Text-driven indoor scene stylization with part-aware details. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1890–1899 (2023)
24. Jin, B., Tian, B., Zhao, H., Zhou, G.: Language-guided semantic style transfer of 3d indoor scenes. In: ACM International Conference on Multimedia 2022 PIES-ME Workshop (2022)
25. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3907–3916 (2018)
26. Kim, B., Azevedo, V.C., Gross, M., Solenthaler, B.: Transport-based neural style transfer for smoke simulations. ACM Transactions on Graphics (TOG) (2019)
27. Kim, G., Kwon, T., Ye, J.C.: Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2426–2435 (2022)
28. Knodt, J., Gao, X.: Consistent mesh diffusion. arXiv preprint arXiv:2312.00971 (2023)
29. Kwon, G., Ye, J.C.: Clipstyler: Image style transfer with a single text condition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18062–18071 (2022)
30. Lee, Y., Kim, K., Kim, H., Sung, M.: Syncdiffusion: Coherent montage via synchronized joint diffusions. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
31. Lei, J., Zhang, Y., Jia, K., et al.: Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. Advances in Neural Information Processing Systems **35**, 30923–30936 (2022)

32. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE transactions on pattern analysis and machine intelligence **30**(2), 228–242 (2007)
33. Liu, H.T.D., Jacobson, A.: Cubic stylization. ACM Transactions on Graphics (2019)
34. Liu, K., Zhan, F., Chen, Y., Zhang, J., Yu, Y., El Saddik, A., Lu, S., Xing, E.P.: Stylerf: Zero-shot 3d style transfer of neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8338–8348 (2023)
35. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4990–4998 (2017)
36. Ma, Y., Zhang, X., Sun, X., Ji, J., Wang, H., Jiang, G., Zhuang, W., Ji, R.: X-mesh: Towards fast and accurate text-driven 3d stylization via dynamic textual guidance. In: roceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
37. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13492–13502 (2022)
38. Müller, T.: tiny-cuda-nn (Apr 2021), `https://github.com/NVlabs/tiny-cuda-nn`
39. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
40. Nguyen-Phuoc, T., Liu, F., Xiao, L.: Snerf: stylized neural implicit representations for 3d scenes. ACM Trans. Graph. (Jul 2022). `https://doi.org/10.1145/3528223.3530107`
41. Pang, H.W., Hua, B.S., Yeung, S.K.: Locally stylized neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 307–316 (2023)
42. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2085–2094 (2021)
43. Pharmapsychotic: CLIP Interrogator (2023), `https://github.com/pharmapsychotic/clip-interrogator`
44. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations (2023)
45. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
46. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501 (2020)
47. Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. In: ACM SIGGRAPH 2023 Conference Proceedings (2023). `https://doi.org/10.1145/3588432.3591503`
48. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)

49. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2007)

50. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)

51. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=St1giarCHLP`

52. Song, L., Cao, L., Xu, H., Kang, K., Tang, F., Yuan, J., Zhao, Y.: Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. In: Proceedings of the 31st ACM International Conference on Multimedia (2023)

53. Sỳkora, D., Jamriška, O., Texler, O., Fišer, J., Lukáč, M., Lu, J., Shechtman, E.: Styleblit: Fast example-based stylization with local guidance. In: Computer Graphics Forum. vol. 38, pp. 83–91. Wiley Online Library (2019)

54. Tumanyan, N., Bar-Tal, O., Amir, S., Bagon, S., Dekel, T.: Disentangling structure and appearance in vit feature space. ACM Trans. Graph. (Nov 2023). `https://doi.org/10.1145/3630096`

55. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: Nerf-art: Text-driven neural radiance fields stylization. IEEE Transactions on Visualization and Computer Graphics (2023)

56. Yang, B., Dong, W., Ma, L., Hu, W., Liu, X., Cui, Z., Ma, Y.: Dreamspace: Dreaming your room space with text-driven panoramic texture propagation. arXiv preprint arXiv:2310.13119 (2023)

57. Yang, H., Chen, Y., Pan, Y., Yao, T., Chen, Z., Mei, T.: 3dstyle-diffusion: Pursuing fine-grained text-driven 3d stylization with 2d diffusion models. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 6860–6868 (2023)

58. Yang, S., Hwang, H., Ye, J.C.: Zero-shot contrastive loss for text-guided diffusion image style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)

59. Yin, K., Gao, J., Shugrina, M., Khamis, S., Fidler, S.: 3dstylenet: Creating 3d shapes with geometric and texture style variations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12456–12465 (2021)

60. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields. In: European Conference on Computer Vision. pp. 717–733. Springer (2022)

61. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3836–3847 (2023)

62. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)

63. Zhang, Y., He, Z., Xing, J., Yao, X., Jia, J.: Ref-npr: Reference-based non-photorealistic radiance fields for controllable scene stylization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4242–4251 (2023)

64. Zhang, Z., Liu, Y., Han, C., Pan, Y., Guo, T., Yao, T.: Transforming radiance field with lipschitz network for photorealistic 3d scene stylization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20712–20721 (2023)