StyleCity: Large-Scale 3D Urban Scenes Stylization – Supplementary Material –

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

Abstract. In this supplementary material, we have more discussions on our system performance in Sec. 1, we provide implementation details of our system in Sec. 2, additional results and discussions in Sec. 3, and more details of quantitative evaluations in Sec. 4.

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

1 More Discussion

While the proposed StyleCity framework has potential, we would like to discuss its limitations and potential future works.

Lack of Real Physics. Supervised by perspective views for 3D stylization, we inevitably misestimate the physical illumination in the 3D world, such as the sunlight direction. Further, our neural texture field merely represents baked color information, while it is able to compass other physical-based properties such as roughness and specularity for more photorealistic rendering. It would be interesting to investigate techniques such as lighting estimation from reference and physically-based 3D stylization using 2D priors.

Sky Synthesis. The synthesized omnidirectional images offer appealing background views while we did not enable any illumination effect on the foreground during neural texture field optimization and rendering. However, it is promising to tailor diffusion models to provide vision-and-text style harmonic environmental light, while how to make the illumination of sky images physically correct is challenging.

Texture Representation and Quality. To enable texture compression for large-scale scene thus easing texture optimization, we adopted a compact neural texture representation employing hash grid encoding. However, with limited network parameters, it may downgrade to some extent visual quality of original texture details with some over-smoothness artifacts particularly for some subtle logos and edges. While increasing network parameters can alleviate the artifacts,

developing a higher-quality but compact texture representation with consumer-level computing resources could be a promising research direction.

Artifacts Derived from Original Texture. The photorealistic style transfer for texture preserves the original content features, relying on high-frequency details, particularly edges. However, when the input textured mesh model contains misaligned and discontinuous textures or broken surfaces, our system tends to generate incorrect stylized texture details. To enhance the final stylization quality, it is potential to explore a pre-processing step that automates the detection and remediation of texture artifacts, ensuring correct semantics and patterns. This research direction holds promise for further advancements in photorealistic style transfer for texture.

2 Implementation Details

2.1 Style-Aligned Sky Synthesis

Our vision-and-text guided panoramic sky synthesis method extends the multi-window joint diffusion technique [2] along with a given image reference as a perceptual similarity constraint and utilizes a novel omnidirectional sampling strategy for high-resolution generation. It successfully binds multiple diffusion generations under equirectangular projections together with shared parameters, while transferring reference-aligned global style to high-resolution seamless panorama synthesis.

Specifically, to generate high-resolution omnidirectional images, we employ the spherical coordinate system to represent 360° sky image and conduct omnidirectional sampling based on Bounding Field of View (BFoV) [11] for joint diffusion. By default, the final output 360° image is in 1024×2048 resolution. We would like to note that using a GPU with larger memory can generate a higher-resolution image. Therefore, we randomly initialize a latent features image $\mathbf{\Omega}$ with a resolution of 128×256. In every denoising step, we sample 90° FoV sub-regions ω on the 360° noisy latent image. The i^{th} sub-region noisy latent feature $\omega^{(i)}$ with a resolution of 64×64 is then sampled from $\mathbf{\Omega}$ through a transformation $\mathcal{T}$:

$$\mathcal{T}_{\mathbf{\Omega} \to \omega^{(i)}} : BFoV_{\mathbf{\Omega}}(clon^{(i)}, clat^{(i)}, \alpha, \beta), \tag{1}$$

where $(clon^{(i)}, clat^{(i)})$ is the longitude and latitude coordinates of the sub-region, while α, β are horizontal and vertical FoVs and set to 90°. Moreover, $clon^{(i)} \in [-\pi, \pi]$, $clat^{(i)} \in [0, \frac{\pi}{4}]$. In total, we extract 64 sub-regions by uniformly sampling along longitude and latitude for each diffusion step.

Before the reverse diffusion step t, we use the style reference image s as the fixed anchor image for perceptual similarity loss $\mathcal{L}_{LPIPS}$ to optimize noisy latent feature $\omega_t^{(i)}$:

$$\hat{\omega}_t^{(i)} : \omega_t^{(i)} - w \cdot \nabla_{\omega_t^{(i)}} \mathcal{L}_{LPIPS}(\mathcal{G}(\Phi(\omega_t^{(i)}, t)), s), \tag{2}$$

where w is the learning rate, ∇ is the gradient, $\mathcal{G}(\cdot)$ is the VAE decoder to predict the final image from a latent feature, $\Phi(\cdot)$ is the DDIM denoising process [15]. Hence, $\mathcal{G}(\Phi(\cdot))$ is the predicted denoised image by decoding the denoised latent feature. The learning rate w is initialized to 20 and exponentially decayed with a 0.95 ratio.

We then apply a joint diffusion process by averaging the weighted latent features of window samples as the full-resolution latent feature image $\hat{\mathbf{\Omega}}$. The denoised latent feature image $\hat{\mathbf{\Omega}}_t$ at step t from the all samples $\hat{\omega}^{(i)}$ is averaged by:

$$\hat{\mathbf{\Omega}}_t = \frac{\sum_i \mathcal{T}^{-1}(\Phi(\hat{\omega}_t^{(i)}, t))}{\sum_i \mathbf{m}_i}, \tag{3}$$

where $\mathbf{m}_i$ is a binary 360° mask for the i^{th} sample region in the 360° image, $\mathcal{T}^{-1}$ is the inverse operation of Eq. 1.

After the diffusion process with 50 steps, we generate the whole 360° sky image with the reference-aligned style by $\mathcal{G}(\hat{\mathbf{\Omega}})$. Different from [13], we use style reference as the fixed anchor image for perceptual similarity loss instead of a generated image as the anchor, which greatly enhances the style consistency. In the main manuscript, Fig. 7a illustrates the overall sky synthesis pipeline, and Fig. 7b displays some visual comparisons with [13]. Sec. 3.1 has more visual comparisons and discussions.

2.2 Pivot-based View Planning Details

An effective camera planning strategy for semantics-aware 3D stylization essentially fulfills two criteria. Firstly, it requires that all planned views can cover comprehensive surfaces with similarly equal possibilities, avoiding unbalanced optimization. Secondly, each single view identifies plausible semantics. We developed a pivot-based view planning method that satisfies the aforementioned criteria. Specifically, to ensure comprehensive sampling of pivot views, we uniformly sample P camera positions on the upper and side faces of the mesh bounding box, with a proper offset distance perpendicular to the closest side. This prevents cameras from being too close to mesh surfaces, enhancing semantics identification. The mesh is then subdivided into r sub-regions based on scene size, where each sub-region is the bounding box of a divided mesh area with a uniform bottom face and a different height. The centroids of these regions are used as camera viewing points and we can obtain $P \times r$ pivot views, locating at P pivot positions and looking at r centroids. During texture optimization, novel training views are sampled along Bezier curves with nearby pivot cameras as control points, and then further augmented by horizontal and vertical translation in the camera space, as depicted in Multi-Scale Progressive Optimization in the main manuscript.

In most experiments, for an urban scene textured model of around $0.5km^2$ and 500MB size, we plan $P = 116$ pivot camera positions and subdivide mesh into $r = 3 \times 3$ parts to calculate 9 centroids, getting a total of 1044 pivot views.

The offset distance is 50 meters. For a larger scene, more pivot views can be planned following the mentioned ratio of the number of views versus area.

2.3 2D-to-3D Segmentation Details

3D model semantic information is required to achieve semantics-aware stylization and avoid style inconsistency among the same categories. Rather than manually annotating a large 3D model, we pre-trained an adapted 2D semantic segmentation model Mask2Former [6] and implemented an automated 2D-to-3D segmentation tool. We found the existing pre-trained Mask2Former models have an unsatisfying performance in urban scenes, such as windows. Therefore, to fine-tune the model we follow the interactive 2D segmentation-and-refinement scheme and iteratively expand annotated data.

Iterative 2D Segmentation Model Fine-tuning and Customization. To achieve 3D segmentation for city scenes with classes of interest ("sky", "building", "window", "road", "plant", "light", "water", "car", "person/animal"), we rely on a pre-trained 2D segmentation model which was fine-tuned with annotated real and synthetic images. We pre-trained an adapted Mask2Former [6] with more network layers with pre-defined classes of interest using the 2D large-scale dataset ADE20k [23]. Then, we predict segment labels for 2D city photos [5], 3D building multi-views [19], and some rendered synthetic city images from collected 3D synthetic models [1], and manually refine labels. These semi-automatically annotated paired data are used to fine-tune the segmentation model. With the increasing accuracy of model segmentation, each iteration workload is reduced. Experimentally, after three iterations with a total of over 8,000 new annotated data, the final segmentation model is qualified to predict semantic segmentation for customized classes. The fine-tuned model is able to segment special views, such as bird's-eye view, and special architectural images, such as rendered synthetic images.

2D-to-3D Segmentation. For a given textured city model, we render 4096×4096 images at the pivot views and predict segments for 16 patches in 1024×1024 for abundant class prediction. Finally, we map segmentation views via UV mapping back to texture image in a low resolution around 3000×3000, which we found sufficient for 3D segmentation and stylization. We adopt a simple mapping strategy, in which we assign a pixel a class value with the highest probability of appearance from all views.

2.4 3D Stylization Training Details

Neural Texture Field. The neural texture field architecture comprises a grid-based hash feature encoding followed by a fully connected network (MLP). The hash grid features are configured with 16 levels of 8-dimension grid features with minimum and maximum resolution of 16^2 and 1024^2 respectively, and the hash

table size is set to 2^{24}. The MLP consists of four linear layers, including three 64-channel layers with ReLU activation, and the final 3-channel layer with Sigmoid activation.

Novel Training View Planning. Given a textured model of around $0.5km^2$ and 500MB size, during the training, we progressively optimize the neural texture field by N=5 levels while the fields of view of sampling view decrease from 90° to 20°. For a larger scene, finer and smaller FoVs should be used accordingly. In each level optimization, we increase one-fold novel views and optimize all sampling views at each level.

Style Losses. To process the input style reference image for scale-adaptive style optimization, we obtain style reference image patches with random cropping and set minimal 4-scale resolutions while a style path long side no smaller than 256 pixels.

Since CLIP vision encoders embed whole image semantics, style-aligned skies are essential for multi-view supervision. We use a random baby blue color as source content sky (in the day), and we project the synthesized sky panorama to the training view as target style sky for CLIP losses.

Optimization. We adopted the Adam optimizer, with betas 0.9 and 0.999. The learning rate is set to $1e-2$ initially and decays by a multiplier of 0.5 every time the optimization level increases. Since dynamically planned views fluctuate, we skip views with few rendered pixels and clip the gradient norm of parameters with a maximum norm of 5.0 to avoid outlier views.

2.5 Photorealism Regularizer

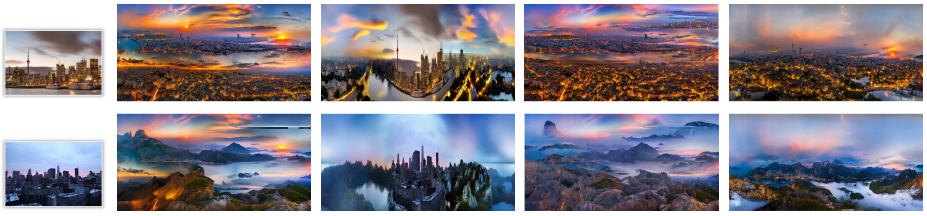
As introduced in the main manuscript, the photorealism regularization term is formulated as:

$$\mathcal{L}_{pht} = \sum_{h=1}^{3} (\mathcal{V}_h^z)^T \mathcal{M}^c \mathcal{V}_h^z, \tag{4}$$

where h denotes a RGB channel, and $\mathcal{V}^z \in \mathbb{R}^{D\times 3} (D = H \times W$, H and W are image resolution) is the flattened version of the stylized view. $\mathcal{M}^c \in \mathbb{R}^{D\times D}$ denotes the matting Laplacian matrix of masked content view, whose (i, j)-th value is:

$$\sum_{k|(i,j)\in w_k} (\delta_{i,j} - \frac{1}{|w_k|} \times (1 + (V_i^c - \mu_k)(\Sigma_k + \frac{\epsilon}{|w_k|} I_3)^{-1} (V_j^c - \mu_k))), \tag{5}$$

where w_k is a k-th 3×3 non-empty window with pixel indices on the content view, $|w_k|$ is the number of pixels in this window, δ and I_3 are respectively $|w_k| \times |w_k|$ and 3×3 identity matrices, $V_i^c \in \mathbb{R}^{3\times 1}$ is i-th pixel value on content view, μ_k is a 3×1 mean vector and Σ_k is 3×3 covariance matrix of colors in w_k, small constant scalar $\epsilon = 1e-7$ by default. We implement it in Pytorch and support CUDA computation.



(a) Style **(b)** MultiDiffusion [2] **(c)** MVDiffusion [16] **(d)** SyncDiffusion [13] **(e)** Ours

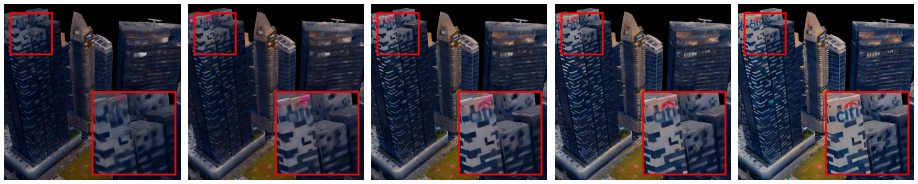
Fig. 1: Comparisons of panoramic sky synthesis with different diffusion-based approaches using the same seeds and text prompts. Zoom-in for better view.

3 Additional Results and Discussions

3.1 Sky Synthesis Comparison

Here, we further compared our style-aligned omnidirectional image synthesis method with other diffusion-based panorama synthesis methods, including MultiDiffusion [2], SyncDiffusion [13], and MVDiffusion [16]. Given the text prompts, MultiDiffusion and SyncDiffusion synthesize panoramas from scratch while MVDiffusion incrementally outpaints the image to generate panoramic images. To run these methods, we used the two text prompts including *"a sky panorama at sunset, a matte painting by Manjit Bawa, Shutterstock contest winner, regionalism, cityscape, creative commons attribution, nightscape."* and *"a sky panorama at dawn, blue hour, a matte painting by Ren Xun, featured on Unsplash, regionalism, photo taken with Ektachrome, nightscape, photo taken with Provia."*.

As the results shown in Fig 1, none of these works can appropriately synthesize panoramas aligning with the desired style. MultiDiffusion only considers multi-window joint diffusion only with different styles in each window. SyncDiffusion has a relatively uniform style among all sampled windows but fails to generate a high-resolution image as an entity. The synthesis images have noticeable vertically layered content. MVDiffusion outpaints existing style image reference with synthesized content while suffering from obviously visible seams like stitching artifacts. Our method synthesizes integrated images that look more consistent omnidirectionally, as demonstrated in Fig. 1e.



(a) Level 1 **(b)** Level 2 **(c)** Level 3 **(d)** Level 4 **(e)** Level 5

Fig. 2: Visualization of different optimization levels. With finer levels of optimization, we learn higher fidelity of stylization.

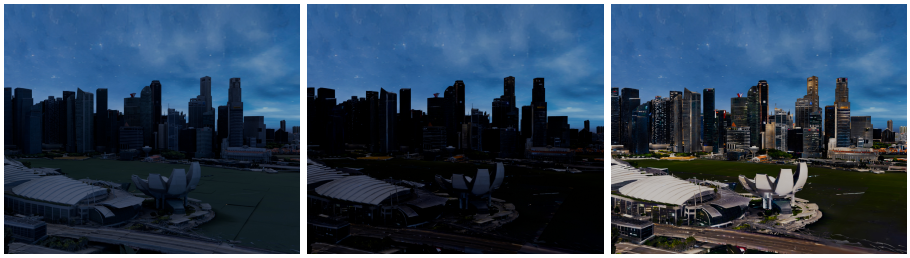
3.2 Effect on Multi-scale Progressive Optimization

We render stylized views with different optimization levels in Fig. 2. With the increasing levels of optimization, our system learns higher-frequency content structure with higher-quality style features, such as clearer logos and lit windows. We optimized the texture with 5 levels by default, and experiments showed that more levels can reach slightly sharper results but may not improve the style a lot.

3.3 Discussion on Environmental Light

To further illustrate the texture baking hallucination of our optimized texture, we present visual comparisons in Fig. 3. The comparisons involve the application of sole ambient lighting on the original texture (Fig.3a) and on our stylized texture (Fig.3b), using our synthesized panoramic sky as the environment map.

Our result (Fig.3c) represents the baking effect in the stylized texture, enabling real-time rendering without the need for ambient or another lighting setup. By contrast, as shown in Fig.3a, the original model with simple ambient light using the synthesized sky as an environment map only shows global color on the model surface, lacking detailed illumination effect on instances. In addition, we introduced ambient light to our stylized model, as depicted in Fig. 3b, which led to the incorrect summation of illumination effect for the final rendition.



(a) + night env map **(b)** Night-time texture + env map **(c)** Ours w/o env map

Fig. 3: Effect of environment light by the synthesized sky at night time style. (a) Original model with ambient light effect or environment map (env map). (b) The stylized night-time model with ambient light effect. (c) Our StyleCity result represents baked texture in practical use with real-time rendering requiring no ambient light effect.

3.4 More Results

Comparison with 2D-Based Approaches. Image or video stylization works downgrade performance in 3D scenarios in terms of 3D geometric and appearance consistency, as discussed in previous 3D stylization works [7, 10, 12]. To further

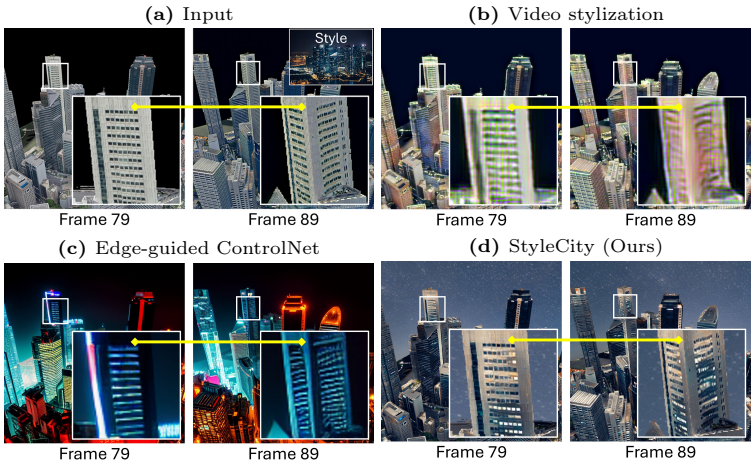


Fig. 4: Comparisons with 2D-based approaches, given the same input of scene and style (image or text) as shown in (a). (b) Photorealistic video style transfer [18]. (c) Edge-guided ControlNet editing for two views given the same random seeds [22]. (d) Our results. Yellow line indicates the identical area. Please zoom in for a better view.

support our argument, we additionally conducted experiments using photorealistic video style transfer by [18] on a sequence, and multi-view editing using edge-guided ControlNet, as depicted in Fig. 4. Results by 2D-based methods lead to obvious multi-view color or structural inconsistency as illustrated in Fig.4b and Fig. 4c, while our results maintain intrinsic 3D consistency.

Besides, 2D stylization does not support stylizing novel views consistently. Video stylization heavily relies on a continuous video trajectory, making it unsuitable for discrete multiple-view inputs.

Multi-view 2D editing by diffusion model conditioned on 3D rendered structure still falls short in realizing sufficient consistency, e.g. Fig. 4c. Moreover, diffusion-based editing suffers from intricate style prompts, as illustrated in comparisons with Instruct-NeRF2NeRF in Fig. 5 and with Text2Tex [3] in Fig. 6.

Baseline Comparisons. Fig. 5 shows more visual comparison results with baselines. Neural field based stylization methods ARF and Instruct-NeRF2NeRF [8, 21] are not robust for diverse scenes and tricky novel-view rendering, leading to blurry artifacts. StyleMesh [9] always synthesizes aliasing artistic patterns, and does not take the background into account. These baselines only focus on global style for a scene with a major central subject, and they cannot achieve semantics-aware local stylization. Urban scene is a special scenario with multiple instances, thus the global optimization approaches cannot achieve plausible results. Our results, by comparison, achieve higher fidelity and more harmonious local-and-global stylization.

Fig. 5: More comparison results with baselines.

(a) Input **(b)** Ours **(c)** Text2Tex [3]

Fig. 6: Comparison with text-guided texture synthesis. (c) Given complex text prompts, text-guided texture synthesis fails to generate reasonable results for the large-scale scene. Inset of (c) is one of the inpainted views for supervision.

StyleCity versus Texture Synthesis. We additionally compare performance with recent text-guided texture synthesis approaches such as Text2Tex [3] in Fig. 6. Different from our texture style transfer problem, they generate a texture from scratch given a mesh without the original texture. We ran Text2Tex's official implementation and used its planned training views for texture painting since our planned views do not aim for texture painting.

Fig. 6 presents a comparison between Text2Tex and our stylization results. When given a city style prompt, Text2Tex falls short in generating high-fidelity texture for large-scale scenes due to limitations imposed by the inpainting diffusion model [22], as depicted in the inset of Fig. 6c. Besides, without original content supervision, Text2Tex fails to preserve place identity. In contrast, our vision-and-text supervised stylization approach achieves a satisfactory stylized texture outcome.

More Stylization Results Fig. 7 and 8 showcase more stylized results of different city models in different time-of-day magic styles and artistic styles.

Fig. 8b displays 2D-to-3D artistic editing results given user input text prompts. We edit a selected view for the same model and input various text prompts for generating text-conditioned edited view references using edge-to-image synthesis [22]. During artistic style optimization, as described in the main manuscript, the texture initialization and the impact of the photorealism regularizer are eliminated to prioritize artistic stroke synthesis, and the weights of the style losses are increased. By emphasizing vision-and-text style features for texture optimization, novel patterns aligned with the given text prompt but unseen in the style image can be generated. We can obtain stylized variations of the same scene by different visual or textual style references, as exemplified in Fig. 8.

4 Quantitative Evaluation Details

4.1 3D Models and 2D References

For quantitative evaluation, we used 5 models, each in around 0.5km $\times$ 0.5km areas, in 5 cities across the world, including London, Sydney, New York, Hong Kong, and Singapore. We selected 5 style images from [5], and generated paired text prompts by [14]; and customized 2 style text prompts including "*a city at night, hyper realistic cyberpunk style, cyberpunk atmosphere, with neon lights and neon signs*", and "*a city at twilight, blue hour, with lights*", and synthesized paired image references by diffusion model [22]. We total used these 7 style references, got total $5 \times 7 = 35$ stylized models for each method for quantitative evaluation. For each model, we render a sequence of 200 testing views around the model.

For qualitative experiments, we collected models from 10 cities including Hong Kong, Singapore, Tokyo (Shibuya City), London, Berlin, Sydney, Los Angeles, New York, Denver and Vancouver.

(a) Input **(b)** Stylized results

Fig. 7: More stylization results of our method.

4.2 Quantitative Metrics

Edge-SSIM Similar to [5, 20], we apply a Canny edge detector for each masked grayscale image and use Canny edge detected maps to compute SSIM [17] between content images and stylized images. The edge-SSIM was found to better represent content similarity regardless of chrominance influence, especially in stylization with dramatic color changes [5].

Masked LPIPS Following [4] evaluation, in each testing view, we use non-sky mask as the foreground mask for masked LPIPS. We use the masked content images and the masked stylized images to get VGG features to compute LPIPS, and then accumulate LPIPS distance on the masked area only.

CLIP Score We use the ViT-L/14 CLIP model to evaluate cosine similarity as CLIP score between stylized testing views and input style text prompts. For fair comparisons, we blended (i.e., copied and pasted) the same synthesized sky background to the same view for all methods.

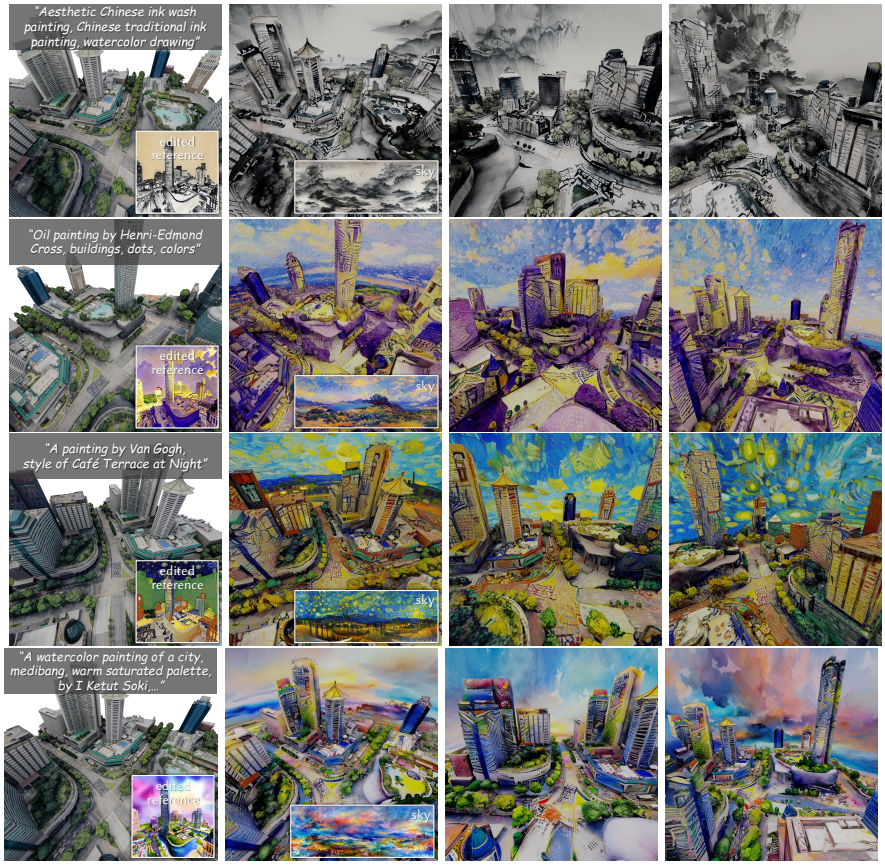
4.3 User Study Details

We conducted a user study using 5 scenes each in a different style. We displayed multi-views with zoom-ins of the original model and stylized models by our method and baselines to participants. Participants are asked to rate a score ranging from 0 to 5 (best) for each stylized model in terms of "content preservation," "style match," and "overall satisfaction." We make the questions as simple as possible in case of participants without enough background knowledge:

1. Content Preservation: *"Does the stylized model look like the original model, and preserve original content structure and identity?"*
2. Style Match: *"Does the style of the model look like the style reference?"*
3. Overall Satisfaction: *"Does the stylized model look good and realistic to you?"*

Before the rating, there is an introduction to the task and a simple exercise of rating by showing an original model and some stylized models. After practicing, participants were shown one stylization case, and then gave scores to each stylized model. In addition, participants were asked to give comments and feedback about why they gave a low score. We got 29 respondents in total, and the final "overall satisfaction" scores are 4.428, 1.588, 2.554, and 2.246 for our method, ARF, InstructN2N, and StyleMesh, respectively.

(a) Hallucination of magic times of a day for a scene.



(b) Artistic editing for a scene given different text prompts.

Fig. 8: More application visual results of our method. (a) Photorealistic stylization with time-of-day effects. (b) Artistic stylization via 2D-to-3D editing given a text prompt; stylized textures tend to gain novel patterns aligned with text prompts.

References

1. Turbosquid (2023), https://www.turbosquid.com/
2. 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)
3. 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)
4. 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)
5. 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)
6. 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)
7. 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)
8. 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)
9. 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)
10. 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)
11. 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)
12. 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)
13. 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)
14. Pharmapsychotic: CLIP Interrogator (2023), https://github.com/pharmapsychotic/clip-interrogator
15. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=St1giarCHLP
16. Tang, S., Zhang, F., Chen, J., Wang, P., Yasutaka, F.: Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)
17. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

18. Wu, Z., Zhu, Z., Du, J., Bai, X.: Ccpl: contrastive coherence preserving loss for versatile style transfer. In: European Conference on Computer Vision. pp. 189–206. Springer (2022)
19. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. Computer Vision and Pattern Recognition (CVPR) (2020)
20. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.W.: Photorealistic style transfer via wavelet transforms. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9036–9045 (2019)
21. 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)
22. 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)
23. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)