3D Photo Stylization: Learning to Generate Stylized Novel Views from a Single Image (Supplementary Material)

We refer to the *supplementary video* on the project webpage[1] for an overview of our results and comparisons with the baselines. This document describes the implementation details of our model, the design of our user study, the details of extending our method to multi-view inputs, a discussion on the choice of depth estimation models, as well as a discussion on the limitation and societal impact of our method.

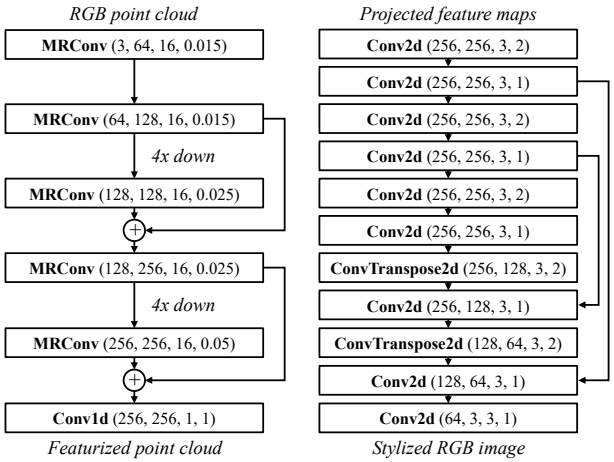
A . Implementation Details

Our model architecture is illustrated in Fig 1. We now present our implementation details.

Point Cloud Encoder Architecture. Our GCN encoder adopts a hierarchical design for computational and memory efficiency. It takes an input RGB point cloud and processes it in three stages with 1, 2 and 2 MRConv layers [5] respectively. The point features are 64, 128 and 256 dimensional after each stage. Contrary to [5], our MRConv variant performs point aggregation using ball queries, and we progressively increase the ball radius throughout the encoder to enlarge its receptive field. At the entry of each stage, we apply farthest point sampling to sub-sample the point cloud by a factor of 4. A residual connection is introduced every two layers to facilitate gradient flow during training. We apply batch normalization [4] after each layer and use ReLU as the non-linearity.

Stylizer Architecture. Our stylizer follows AdaAttN [7]. We apply AdaAttN once on the relu3_1 features of VGG to modulate the GCN output since our GCN architecture loosely mirrors the first three stages of VGG. As discussed in the main paper, we apply a multi-layer perceptron (MLP) with two fully-connected layers of 256 units to map content features to the style feature space before stylization. A symmetric MLP is applied after stylization to bring the modulated features back to the content feature space. The MLPs use ReLU as the non-linearity.

Neural Renderer Architecture. Our neural renderer first up-samples the 256-dimensional encoder output via inverse distance weighted interpolation [9] until the output resolution is the same as the encoder input. Following [9], we set the fall-off coefficient to 2 and the number of neighbors to 3 in up-sampling. The rasterizer [8] projects the up-sampled point features to the image plane of a novel view given camera pose and intrinsics. The resulting 2D feature maps have 256 dimensions and are further processed by a U-Net [12] with three levels. The encoder part of the U-Net down-samples the feature maps *without inflating the channel dimension*. We interpret this as a learnable anti-aliasing step in the same spirit as widely used super-sampling in computer graphics. The decoder part subsequently up-samples the feature maps via transposed convolution and meanwhile halves the channel dimension. The skip connections, implemented as 1×1 convs, pass along feature from the encoder to the decoder to facilitate gradient flow. All layers in the U-Net except the skip convs have a kernel size of 3×3. We apply leaky ReLU with a slope of 0.2 in the encoder and ReLU in the decoder as the non-linearity.



(a) Point cloud encoder **(b) Neural renderer**

Figure 1. **Model architecture.** Architecture of our point cloud encoder and neural renderer. The layer specifications are as follows: **Conv1/2d** (input channel, output channel, kernel size, stride); **MRConv** (input channel, output channel, maximum number of neighboring points, ball radius).

[1]Project page: http://pages.cs.wisc.edu/~fmu/style3d

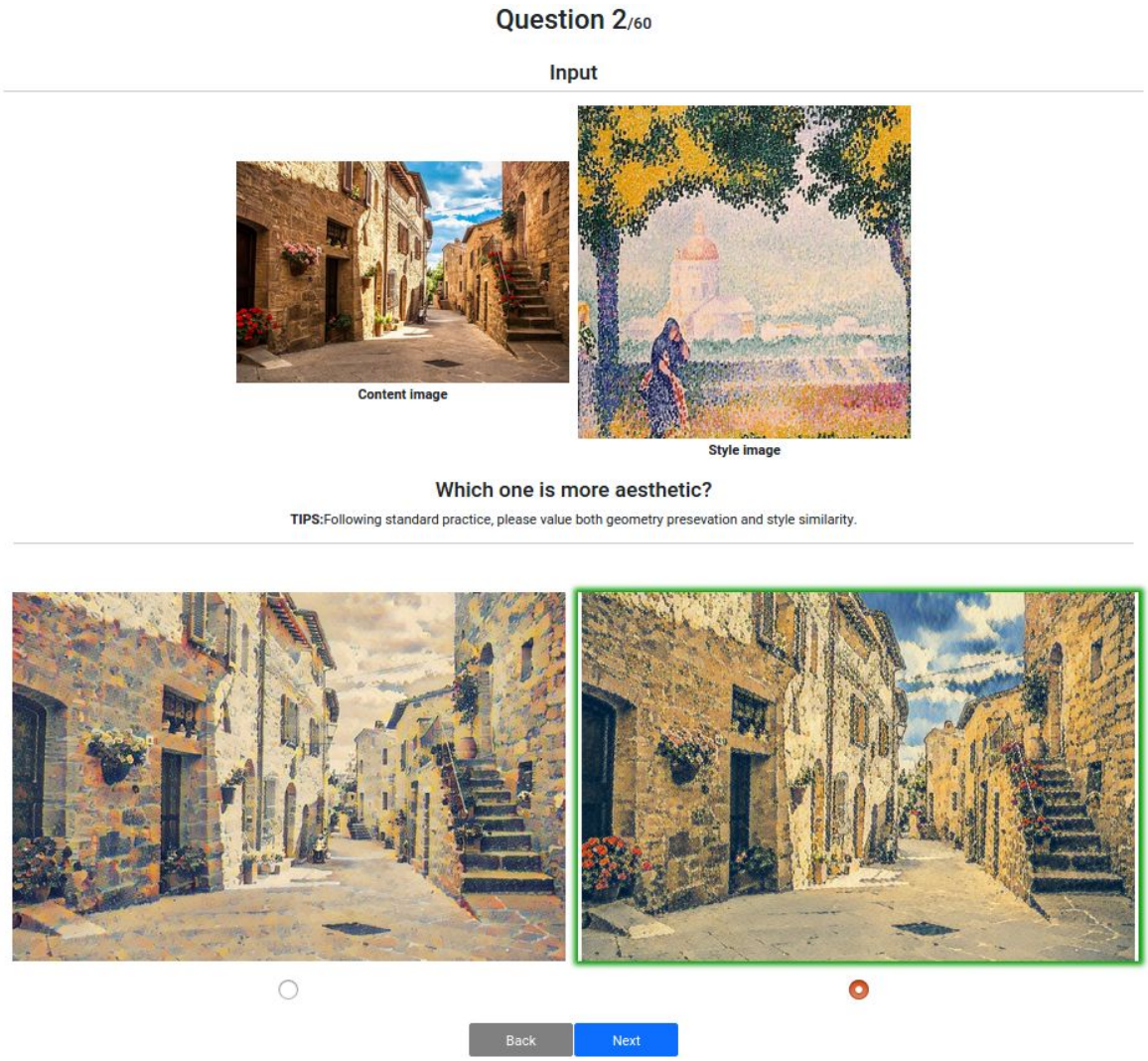


Figure 2. **Screenshot of our user study.** A randomly picked question in our user study.

B . User Study Design

We conduct a user study to compare our method with baselines that sequentially combine 3DPhoto [13] and one of the six image [3, 6, 7] or video style transfer methods [1, 7, 14]. The study includes three sections for the assessment of style quality, multi-view consistency and overall synthesis quality. Each section consists of 60 random binary choice questions that compare our method with one of the baselines. For convenience, a stylized 3D photo is displayed as a 90-frame snippet following a random camera trajectory. For fair evaluation of style quality, we only display stylized image of the input view so as not to bias participants toward more consistent renderings. Similarly, we hide the content and style images when consistency is evaluated to minimize the impact of style quality. Our analysis is based on a total of 5,400 votes collected from 30 volunteers. We show a screenshot of our user study in Fig 2. Our user study is anonymous and does not involve the collection of personally identifiable data.

C . Details on Extension to Multi-view Inputs

Extending our method to the multi-view setting is immediate after a small modification on point cloud normalization. Now that more than one input views are available, we back-project all views to a point cloud and transform it into the NDC space anchored to the center view. Everything else stays exactly the same, and importantly, the model need not be re-trained thanks to the normalization step. In our experiments, we use the same depth maps from [2] for a fair comparison with StyleScene [2]. Those results were shown in Table 3 and Figure 10 of our main paper.

D . Choice of Depth estimation model

We employ LeReS [15] as the depth estimator at training time. LeReS can infer scene scale and camera field of view from an input image. We use it in training to ensure the plausibility of the synthesized 3D data. Importantly, one may drop in any depth estimation model at inference time without re-training other model components thanks to our proposed point cloud normalization technique. This allows one to explore the complementary strength of off-the-shelf depth estimation models on different scene categories or under varying resource constraints. We present stylization results of the same input image under three state-of-the-art depth estimation models (DPT [10], LeReS [15] and MiDaS [11]) in Fig 3. Note that depth maps produced by different methods result in slight variation in scene coverage given the same camera pose.

E . Limitations

Despite steady progress in monocular depth estimation, current state of the arts do not always produce reliable depth maps for complex scenes, and in particular for those pixels near depth discontinuities. Our method relies on monocular depth estimation on the input image and thus inherits the failure mode of the underlying depth estimators (Fig 3). As a partial remedy, we have demonstrated an extension of our method to use mutli-view inputs with more reliable depth estimations. Another limitation our method shares with StyleScene [2] lies in the run-time speed. While our method renders stylized images of 1K resolution at interactive rate on a TITAN Xp GPU, the current implementation may not support interactive exploration of a high-resolution stylized 3D photo on mobile devices. Future work may focus on improving rendering speed for 3D photo stylization.

Societal impacts: We anticipate that our research would facilitate new applications of 3D content creation from 2D photos. Similar to other image manipulation methods like neural style transfer, our method might face potential copyright infringement, when copyright-protected content images are modified and improperly used.

References

[1] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. In *AAAI*, 2021. 2

[2] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *ICCV*, 2021. 2, 3

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1

[5] Guohao Li, Matthias Müller, Guocheng Qian, Itzel Carolina Delgadillo Perez, Abdulellah Abualshour, Ali Kassem Thabet, and Bernard Ghanem. Deepgcns: Making gcns go as deep as cnns. *TPAMI*, 2021. 1

[6] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *CVPR*, 2019. 2

[7] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *ICCV*, 2021. 1, 2

[8] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *TOG*, 2019. 1

[9] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017. 1

[10] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. 2021. 3, 4

[11] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020. 3, 4

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[13] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020. 2

[14] Wenjing Wang, Jizheng Xu, Li Zhang, Yue Wang, and Jiaying Liu. Consistent video style transfer via compound regularization. In *AAAI*, 2020. 2

[15] Wei Yin, Jianming Zhang, Oliver Wang, Simon Niklaus, Long Mai, Simon Chen, and Chunhua Shen. Learning to recover 3d scene shape from a single image. In *CVPR*, 2021. 3, 4

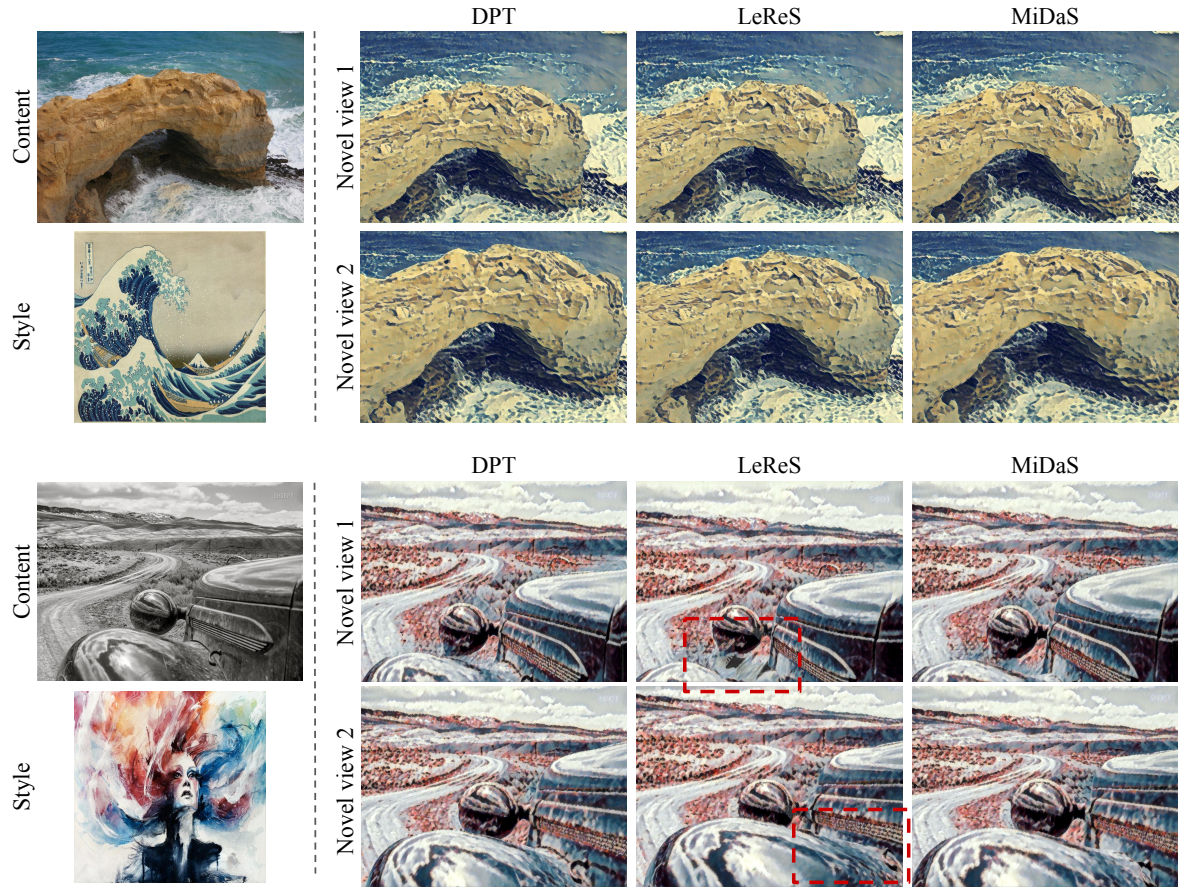


Figure 3. **Stylization results with different depth estimation models.** We employ LeReS [15] as the depth estimator at training time and show that one may drop in any depth estimation method (DPT [10], LeReS [15] or MiDaS [11]) at inference time without re-training. One limitation of our method is that it is susceptible to error in depth estimation (red boxes).